518 Rec'd PCT/PTO 2 5 JUL 2001

FORM PTO-1390 (Modified) (REV 11-98) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371	112740-254
	U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR 09/889890

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/EP00/00317	**17 January 2000**	**21 January 1999**

TITLE OF INVENTION

LOAD DISTRIBUTION METHOD OF A MULTIPROCESSOR SYSTEM, AND MULTIPROCESSOR SYSTEM

APPLICANT(S) FOR DO/EO/US

Peter Hanselka et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
   - a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   - b. ☐ has been transmitted by the International Bureau.
   - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
   - a. ☒ are transmitted herewith (required only if not transmitted by the International Bureau).
   - b. ☐ have been transmitted by the International Bureau.
   - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   - d. ☐ have not been made and will not be made.
9. ☒ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☒ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 13 to 20 below concern document(s) or information included:

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☒ A substitute specification.
18. ☐ A change of power of attorney and/or address letter.
19. ☒ Certificate of Mailing by Express Mail
20. ☒ Other items or information:

Submission of Drawings FIgures 1-3 on three sheets

JC17 Rec'd PCT/PTO 23 JUL 2001

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
09/889890	PCT/EP00/00317	112740-254

21. The following fees are submitted:.

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

		CALCULATIONS	PTO USE ONLY
☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2) paid to USPTO and International Search Report not prepared by the EPO or JPO	$1,000.00		
☒ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but Internation Search Report prepared by the EPO or JPO	$860.00		
☐ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO	$710.00		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4)	$690.00		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4)	$100.00		
ENTER APPROPRIATE BASIC FEE AMOUNT =		$860.00	
Surcharge of **$130.00** for furnishing the oath or declaration later than months from the earliest claimed priority date (37 CFR 1.492 (e)).	☐ 20 ☐ 30	$0.00	

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	35 - 20 =	15	x $18.00	$270.00	
Independent claims	2 - 3 =	0	x $80.00	$0.00	
Multiple Dependent Claims **(check if applicable).**			☐	$0.00	
	TOTAL OF ABOVE CALCULATIONS		=	$1,130.00	
Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) **(check if applicable).**			☐	$0.00	
	SUBTOTAL		=	$1,130.00	
Processing fee of **$130.00** for furnishing the English translation later than months from the earliest claimed priority date (37 CFR 1.492 (f)).		☐ 20 ☐ 30	+	$0.00	
	TOTAL NATIONAL FEE		=	$1,130.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) **(check if applicable).**			☐	$0.00	
	TOTAL FEES ENCLOSED		=	$1,130.00	
				Amount to be: refunded	$
				charged	$

☒ A check in the amount of **$1,130.00** to cover the above fees is enclosed.

☐ Please charge my Deposit Account No. in the amount of to cover the above fees. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **02-1818** A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

William E. Vaughan (Reg. No. 39,056)
Bell, Boyd & Lloyd LLC
P.O. Box 1135
Chicago, Illinois 60690

SIGNATURE

William E. Vaughan
NAME

39,056
REGISTRATION NUMBER

July 23, 2001
DATE

BOX PCT

IN THE UNITED STATES ELECTED/DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

PRELIMINARY AMENDMENT

APPLICANTS:	Peter Hanselka et al.	DOCKET NO: 112740-254
SERIAL NO:		GROUP ART UNIT:
		EXAMINER:
INTERNATIONAL APPLICATION NO:		PCT/EP00/00317
INTERNATIONAL FILING DATE:		17 January 2000
INVENTION:	LOAD DISTRIBUTION METHOD OF A MULTIPROCESSOR SYSTEM, AND MULTIPROCESSOR SYSTEM	

Assistant Commissioner for Patents,
Washington, D.C. 20231

Sir:

Please amend the above-identified International Application before entry into the National stage before the U.S. Patent and Trademark Office under 35 U.S.C. §371 as follows:

In the Specification:

Please replace the Specification of the present application, including the Abstract, with the following Substitute Specification:

SPECIFICATION

TITLE

LOAD DISTRIBUTION METHOD OF A MULTIPROCESSOR SYSTEM, AND MULTIPROCESSOR SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a number of processors MP_i

(where $i=1,2,...,n$) under real-time conditions. The present invention further relates to a multiprocessor system, in particular of a communication system, having a load distribution mechanism.

Description of the Prior Art

A similar method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, is disclosed, for example, in the applicant's European patent application EP 0 645 702 A1. This document discloses a method for load balancing in a multiprocessor system, in particular a multiprocessor system of a communication system, in which tasks that arise can be processed by a number of processors under real time conditions, in which case, in order to perform the load balancing, the following method steps are mentioned:

- each processor determines its load state in the form of a quantified magnitude,
- the load states of the other processors are communicated to each processor within a time frame,
- depending on its load state exceeding a specific magnitude and depending on the load states of the remaining processors, each processor outputs at least a portion of the tasks arising in it to the remaining processors, and
- the output tasks are divided between the remaining processors in accordance with the load states thereof.

In the example, the method is concretized to the effect that distribution quotas are calculated during operation continually and before entry into the load distribution, which, in this case, does not begin until after a specific overload has been reached, according to which distribution quotas the individual processors output their distributable load to other processors in the case of overload. If the system is permanently utilized to capacity in a nonuniform manner, then the load is distributed only in the event of overload of one or more processors. This is accompanied by unnecessary load rejection, however. Reducing the overload threshold to a lower value does not lead to a satisfactory result because an

unnecessarily large amount of load is distributed and oscillation states can arise. This situation emerges from the assumption made there that the overload or the nonuniform loading lasts for a short duration.

Furthermore, reference is made to Evans D.J. et al., "Dynamic Load Balancing using Task Transfer Probabilities" in Parallel Computing, Vol. 19, No. 8, August 1, 1993, pages 897-916, which also presents a load distribution strategy taking account of load distribution probabilities, although exclusively currently measured values are taken into consideration. The temporal development of the parameters considered is not taken into account.

It is an object of the invention, therefore, to specify an improved load distribution method for a multiprocessor system which commences in good time and in a "soft" fashion and thereby eliminates permanent unbalanced load states in the load offer without load rejection. Moreover, the intention is also to specify a corresponding multiprocessor system.

SUMMARY OF THE INVENTION

Accordingly, the inventors propose a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a number of processors MP_i (where $i = 1,2,...,n$) under real-time conditions, having the following iterative method steps that are repeated at time intervals CI:

- each processor MP_i determines its actual load state Y_i and estimates as a function of previously communicated distribution quotas $q_i(old)$ (where q_i = load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,
- each processor MP_i indirectly or directly communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where $k = 1,2,...i-1,i+1,...n$),
- each processor MP_i determines its load distribution factors p_{ij} (where $j = 1,2,...n$) as a function of the load indication values $MPbi_k$ of the other processors MP_k,

- each processor MP_i determines its distribution quota $q_i(new)$ as a function of its actual load state Y_i and the load distribution factors P_{ij},

- on the basis of its quota q_i and its load distribution factors p_{ij}, each processor MP_i distributes its distributable load to other processors MP_k if its distribution quota $q_i(new)$ exceeds a predetermined value q_v.

In order to estimate the offered load A_i of a processor MP_i, it is advantageous to use the formula $A_i{:=}Y_i/(1-q_iV)$. A_i and Y_i can be specified in units of erlangs, while the variables q_i and V are dimensionless fraction indications in accordance with their meaning.

It is also advantageous to subdivide the multi-value load indication value (balancing indicator) $MPbi_i$ into three discrete values, preferably the following demarcation with threshold values holding true: NORMAL for $MPbi_i$ if the processor capacity utilization is from 0 to 70%, HIGH for $MPbi_i$ if the processor capacity utilization is from 70% to 85%, and OVERLOAD for $MPbi_i$ if the processor capacity utilization is above 85%.

It is also advantageous if a hysteresis is introduced in the case of a load state alteration on account of threshold value overshooting or threshold value undershooting in the case of rising or falling processor capacity utilization.

Moreover, it may be advantageous if the load indication value (balancing indicator) $MPbi_i$ is subject to a temporal hysteresis with regard to changes and thus experiences a certain inertia. Values of 1 to 2 time intervals CI can be assumed as hysteresis limit.

Further advantageous assumptions in the performance of the method according to the present invention are: the typical distributable proportion V of a typical task shall be the average or maximum proportion, and an average or maximum processing time of a task shall be assumed as the typical processing time of a task. In this case, the respective average value or maximum value of a proportion and respectively of a task also can be continually determined during the operating time and, if appropriate, be incorporated as a moving value and be adopted in updated form into the load distribution method. It is favorable here if the

time duration over which the moving values are determined is long relative to the control interval CI.

It is also particularly advantageous if the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.05<q_V<0.3$, preferably $0.1<q_v<0.25$, preferably $q_v=0.2$.

Furthermore, the method according to the present invention can be configured particularly advantageously if the following criteria are satisfied in the calculation of the distribution quota q_i:

- $p_{ii}: = 0$
- if $MPbi_j$ corresponds to an average load, preferably $MPbi_j$ = NORMAL, the following holds true:

$p_{ij}(new) = pij(old)+p_{c1}/n$, for $j=1,...,n$ and $i\neq j$

- if $MPbi_j$ corresponds to a high load, preferably $MPbi_j$=HIGH, the following holds true:

$p_{ij}(new) = p_{ij}(old)-p_{c2}/n$, for $j=1,...,n$ and $i\neq j$

- if $MPbi_j$ corresponds to an overload, preferably $MPbi_j$=OVERLOAD, the following holds true:

$p_{ij}(new) = 0$

- in which case the $p_{ij\ (j=1,...,n)}$ is normalized to 1 with the sum p_{sum} of the p_{ij} and
- as initialization value at the beginning of the distribution processes, all p_{ij}, excluding p_{ii}, are identical.

As advantageous numerical values, $0.1<p_{c1}<0.5$, preferably $0.2<p_{c1}<0.3$ and preferably $p_{c1}=0.25$ may be assumed for the constant p_{c1}. Equally, it is advantageous to set $0.1<p_{c2}<0.5$, preferably $0.2<p_{c2}<0.3$, preferably $p_{c2}=0.25$ for the constant p_{c2}. Moreover, the initialization value of the p_{ij} at the beginning of the distribution processes can be set to be equal to $(n-1)^{-1}$.

Furthermore, the method according to the present invention can be configured particularly advantageously if each processor Mp_i determines a multi-

value load status (load state) $MP1s_i$ on the basis of its actual current load Y_i, and the following criteria are satisfied in the calculation of the load indication values $MPbi_i$:

- if $MPls_i$ corresponds to the highest load, preferably $MPls_i$=EXTREME, the following holds true:

$q_i(new)=c_{q1}$,

- if $p_{sum} \geq 1$ holds true:

- if the actual load state Y_i is greater than a predetermined value $threshold_H$, q_i is increased where $q_i=\min\{q_i+c_{q1},1\}$,

- if the actual load state Y_i is less than a predetermined value $threshold_N$, q_i is decreased where $q_i=\max\{q_i-c_{q2}, c_{q3}\}$, where $0<c_{q3}<q_v$, preferably $c_{q3}=0.1$,

- otherwise ($threshold_N \leq Y_i \leq threshold_H$), qi obtains an intermediate value between the two alternatives mentioned above, preferably by linear interpolation

- if $p_{sum} \leq 1$ holds true: $q_i(new)=q_i(old) * p_{sum}$.

With regard to the multi-value load status (load state) $MPls_i$, the assumption of four discrete values is proposed as being particularly preferred, the following preferably being assumed: NORMAL for $MPls_i$ if the processor capacity utilization lies below 70%, HIGH for $MPls_i$ if the processor capacity utilization is from 70% to 85%, OVERLAND for $MPls_i$ if the processor capacity utilization lies above 85%, and EXTREME for $MPls_i$ if the load state OVERLAND permanently prevails. In this case, two, is maybe advantageous if the load status (load state) $MPls_i$ is subject to a hysteresis with regard to changes. Values of 1 to 2 time intervals CI advantageously can be assumed as hysteresis limit.

For optimal configuration of the method of the present invention, the following ranges of numbers of numerical values are preferred for the constant c_{q1}: $0.05<c_{q1}<0.3$, preferably $0.1<c_{q1}<0.2$, preferably $c_{q1}=0.15$. Moreover, preferably $0.05<c_{q2}<0.2$, preferably $c_{q2}=0.10$, can be assumed for the constant c_{q2}.

With regard to the constant $threshold_N$, the following is regarded as a preferred range of values: $0.6< threshold_N <0.8$, preferably $threshold_N=0.7$.

With regard to the constant $threshold_H$, the following is regarded as a preferred range of values: 0.7< $threshold_H$ <0.95, preferably $threshold_H$=0.85.

Another configuration of the method according to the present invention provides for an overload value OL_i of the processors MP_i to be additionally determined in each time interval CI, which value is a measure of the magnitude of the overload and serves as a benchmark for overload rejection, where OL_i=0,1,...m, OL_i representing a quantification for the overload of the processor, and the distribution quota q_i to be increased in any case if OL_i>0 where $q_i(new):=\min\{qi(old)+c_{q1},1\}$.

According to the present invention, it is also possible to adapt the load distribution method to changing boundary conditions by the above-specified constants (q_v, p_{c1}, p_{c2}, q_{c1}, q_{c2}, $threshold_N$, $threshold_N$, c_{q1}, c_{q2}, c_{q3}) being at least partly adapted during operation.

The present invention additionally proposes a multiprocessor system, in particular of a communication system, having a number of processors MP_i (where i=1,2,...,n) for executing tasks that arise under real-time conditions, in which case:

- each processor MP_i determines its actual load state Y_i, and estimates as a function of previously communicated distribution quotas $q_i(old)$ (where q_i=load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,

- each processor MP_i indirectly or directly communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where k = 1,2,...i-1,i+1,...n),

- each processor MP_i determines its load distribution probabilities p_{ij} (where j = 1,2,...n) as a function of the load indication values $MPbi_k$ of the other processors MP_k,

- each processor MP_i determines its distribution quota $q_i(new)$ as a function of its actual load state Y_i and

- each processor MP_i distributes, on the basis of its quota q_i and its load distribution factors p_{ij},

its distributable load to other processors MP_k if its distribution quota q_i(new) exceeds a predetermined value q_v.

According to the present invention, the multiprocessor system proposed above can be configured such that one of the abovementioned methods is implemented in each case, the implementation being effected by corresponding programming of the processors.

It should also be pointed out that the index (old) relates, in each case, to the values of the preceding iteration step, and the index (new) relates to the now current iteration step.

The particular advantage of the method according to the present invention and of the corresponding multiprocessor system is that, in contrast to the prior art mentioned in the introduction, it ensures a "soft" entry into the load distribution and, as a result, is more adaptable and less susceptible to unbalanced load situations wherein oscillation states are better avoided. Ultimately, this reduces the probability of the rejection of tasks, in particular switching tasks.

Additional features and advantages of the present invention are described in, and will be apparent from, the following Detailed Description of the Preferred Embodiments and the Drawings.

DESCRIPTION OF THE DRAWINGS

Figure 1 shows a flow diagram of an arising and distributed load offer;

Figure 2a shows a graphical illustration of the decisions for updating the load distribution factors p_{ij};

Figure 2b shows a graphical illustration of the decisions for updating the distribution quotas q_I; and

Figure 3 shows a formula for linear interpolation of q_i.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The method according to the present invention (normal load balancing=NLB) is a load balancing method in which quotas are set and which runs on a multiprocessor system, in particular in a switching center of a communication system, for distributing operating loads that arise between the respective other

processors. It is intended to ensure that lengthy unbalanced load situations are eliminated, and as far as possible, all requested tasks are processed in the shortest possible time. A particularly advantageous embodiment of this method will be described below.

Each processor MP_i where $i=1,2,...,n$ carries a distribution quota q_i, which fixes the proportion V of the distributable load which is actually to be distributed. Such a quota enables a softer entry or exit from the load distribution to other processors. Oscillation states and load fluctuations are avoided in this way. This may be the case, for example, if a processor distributes so much load to another processor that the latter is in turn overloaded.

The distribution quota q_i is determined anew at each time interval CI. The only information required by the other processors MP_k where $k=1,...i-1,i+1,...n$ for each CI are load value indicators (balancing indicators) $MPbi_i$. These load value indicators are, similarly to the load status values (load states)from the load control, load states having the significances NORMAL, HIGH or OVERLOAD. While the load state is determined on the basis of the actually processed load Y_i of the processor MP_i, the load value indicator $MPbi_i$ is determined from an estimation of the currently offered load A_i. The estimated offered load A_i may, due to load distribution, be considerably more than the actually processed load Y_i and constitutes the crucial quantity which (in the form of the load value indicator $MPbi_i$) is made available as information by one processor MP_i to the others MP_k.

In addition to the distribution quota q_i, each MP_i carries probabilities p_{ij} which indicate the probability that, in the event of load distribution, load will be transferred from the i-th processor MP_i to the j-th processor MP_j. The probabilities are determined in such a way that if, for instance, the j-th processor MP_j already has a large amount of load to process and, therefore, can only take up a small amount of additional load, the associated p_{ij} is less than the P_{ik} for a free MP_k.

Figure 1 illustrates the interaction of the p_{ij} and q_i. The double indexing “ij” of the characteristic quantities refers to the respective processor with the number of the first index (here i) in each case knowing a “column” of n values with the second

index (here j). It should be noted that each processor only knows its relevant values (that is to say its column), overall a square matrix being known in the system. Thus, for example, p_{ij} is the probability that load will be distributed from the i-th MP to the j-th MP if the i-th MP has too much load.

In Figure 1, moreover, the actually processed load of the j-th processor MP_j is designated by Y_j, the estimated offered load is designated by A_j and that part of the load offer which can be shifted is designated by a. The load situation shown is overload (OVERLOAD) on MP_1, there still being space for additional tasks on the MP_k where k=2,3,4. Figure 1 shows how the MP_1 processes a first part of the load itself and distributes the remainder a. Of this remainder a, the largest proportion goes to MP_3 and the smallest proportion to MP_4, which, in this example, already has a large amount of its own load to process. The loads which the MP_k additionally receive besides that from MP_1 are not depicted. The width of the flow bars represents a measure of the magnitude of the load.

The following algorithm is thus produced in accordance with the concept of the present invention: if the j-th processor MP_j reports the balancing indicator NORMAL, the p_{ij} is increased on the MP_i respectively considered. The probability that this processor MP_i will output load to MP_j if it has to distribute load thus rises. If the balancing indicator HIGH is reported, then the p_{ij} is decreased. If the balancing indicator OVERLOAD is reported, p_{ij} is set to zero, with the result that no load is output to the j-th processor MP_j. The distribution quota q_i is changed following the determination of the p_{ij}. If many of the p_{ij} were able to be increased, then the sum of the p_{ij} over j is greater than 1 and there is evidently still space on the other processors MP_k. The distribution quota q_i can, thus, be changed according to the requirements of the processor (considered).
The distribution quota q_i is increased in the event of high load Y_i on the processor MP_i considered, and q_i is decreased in the event of low load. If many of the p_{ij} have been reduced, then the sum of the p_{ij} over j is less than 1 and the distribution quota q_i must be reduced.

An illustration of these decisions is represented in Figures 2 and 2b. The decision diagrams show the updating algorithms for p_{ij} (Figure 2a) and for distribution quota q_i (Figure 2b), which are carried out in each time interval CI for the i-th processor MP_i.

In the load distribution method (NLB) according to the present invention, some parameters (constants) are required, the choice of which can greatly influence the behavior in specific load situations. In most cases this results in a conflict between a load distribution method which can react rapidly to load changes, and a stable load distribution method which does not tend toward oscillations and further distribution of tasks. In this case, "further distribution" refers to the simultaneous distribution of dedicated load and the processing of extrinsic load on one processor.

The following parameter changes bring about a more rapidly reacting NLB:

- the relatively great alteration of q_i where: $0.15<c_{q1}$, $0.1<c_{q2}$
- the relatively great alteration of the p_{ij} where: $0.25<p_{c1}$, $0.25<p_{c2}$
- the relatively late setting of the load indication values $MPbi_1$ where: $threshold_H>0.7$ (i.e. report only in the event of relatively high load 'HIGH') to the other processors MPk.

In detail, the preferred method thus proceeds as follows in a multiprocessor communication computer:

As the duration of the time interval (control interval) CI of the time frame with which the method iteratively proceeds, 1 to 2 seconds is preferably chosen in the case of the presently known multiprocessor systems appertaining to switching technology. It goes without saying that the time interval can be shortened with rising processor power.

The quantities q_i, p_{ij}, $MPls_i$ and MPb_{ij} are updated in each control interval CI.

The actually processed load Y_i of a processor MP_i is determined as processor run time quantity, measured in erlangs.

The estimated offered load A_j of a processor MP_i is determined from the distribution quota q_i of the current control interval CI and the estimated distributable proportion of an average task, for example the processing of a call.

The following holds true:

The number of processors MP_i in the multiprocessor system is n.

$A_i := Y_i/(1-q_iV)$, where V is the distributable proportion of a call.

$MPls_i$: load state of the i-th MP, can assume the values NORMAL, HIGH, OVERLOAD or EXTREME. The actually processed load Yi is used to calculate the load state.

In order to avoid premature changes of the $MPls_i$, hystereses are introduced. If, for instance, the $MPls_i$ is set from NORMAL to HIGH, it must be the case that $Y_i > threshold_N + \Delta_+$, whereas, in order to get from HIGH to NORMAL, it must be the case that $Y_i < threshold_N - \Delta_-$. This procedure is also known as the high water-low water method. In the case of EXTREME, the distribution method (load balancing) must be switched off for this processor MP_i, for system engineering reasons relating to the switching center.

$threshold_N$: is the normal load threshold - after taking a hysteresis into account, the MPls is recorded as NORMAL below the said threshold and as HIGH above said threshold.

$threshold_H$: High load threshold - after taking a hysteresis and a load-dependent temporal delay (start indicator) into account, the MPls is recorded as HIGH below this threshold and as OVERLOAD above said threshold.

The load indication value (balancing indicator) $MPbi_i$ of the i-th processor MP_i can assume the values NORMAL, HIGH or OVERLOAD. This value is calculated like the $MPls_i$, except that here, instead of the actual load Y_i, the estimated offered load A_i is taken as a basis and other values are adopted for Δ_+ and Δ_-, where $\Delta_+ = \Delta_- = 0.02$.

In addition, an Overload Level OL_i of the processor MP_i is determined, which can assume the values 0... 6 and is conceived as quantification of the

overload state of the processor MP_i. If the OL_i>0, calls are rejected; the higher the value, the greater the probability that a call will be rejected.

The load which is to be distributed from MP_i to MP_j is expressed as a probability p_{ij} and can, thus, assume values between 0 and 1.

The magnitude of the value p_{ij} is determined by the following criteria:

- initialize p_{ij} where $p_{ij}:=(n-1)^{-1}$
- p_{ii}: 0, MP_i should not distribute to itself.
- If $MPbi_j$=NORMAL: $p_{ij} \rightarrow p_{ij} + 0.25/n$, $j=1,...,n$, $i \neq j$. The old p_{ij} can be increased because there is still space on the processor MP_j.
- If $MPbi_j$=HIGH: $p_{ij} \rightarrow p_{ij} - 0.25/n$. The old p_{ij} must be decreased because MP_j is utilized to full capacity.
- If $MPbi_j$=OVERLOAD: $p_{ij} = 0$. No load should be output to overloaded processors MP_n.

The newly determined p_{ij} must still be normalized:

Set p_{sum} = sum (p_{ij}) over $j=1,...,n$ and normalize (if $p_{sum} > 0$) where $p_{ij} \rightarrow p_{ij}/p_{sum}$

Afterward, the distribution quota q_i is determined using the following criteria:

- Initialization value: $q_i = 0.1$
- If the $MPls_i$ =EXTREME: $q_i = 0.1$. This MP is overloaded so severely that even its own proportion for a distributed call would overtax it. Therefore, no load balancing, rather only rejecting; load balancing is not practical, moreover, for system engineering reasons relating to the switching center.
- If $p_{sum} > 1$, more load can evidently be distributed. q_i can then be determined according to the requirements of the MP_i, where:

1. If the $OL_i > 0$, increase q_i in any case, where: $q_i \rightarrow \min \{q_i + 0.15, 1\}$
2. If $Y_i > threshold_H$, increase q_i, where: $q_i \rightarrow \min \{q_i + 0.15, 1\}$
3. If $Y_i < threshold_N$, decrease q_i, where: $q_i \rightarrow \max \{q_i - 0.10, 0.1\}$
4. Otherwise, if $threshold_N < Y_i < threshold_H$ the following holds true:

$q_i \rightarrow \min\{\max\{q_i+(0.25/\text{threshold}_H\text{-threshold}_N)) * (Y_i\text{-threshold}_N)\text{-}0.1, 0.1\}, 1.0\}$.

This is the linear interpolation between the above increase by 0.15 and the above decrease by 0.1. The formula is represented again more readably in Figure 3.

- If $p_{sum} < 1$, evidently too much load was distributed and q_i must be decreased, where: $q_i \rightarrow q_i * p_{sum}$.

- The processor MP_i distributes load to other processors MP_k if it becomes the case that $q_i > 0.25$.

The method according to the present invention thus has the following properties and advantages:

A very small information overhead exists between the processors participating in the load distribution method. Only a few, preferably three-value, load states are reciprocally known, which load states are updated and distributed only once per control interval.

For each processor, there is a quota which is updated in each control interval and regulates the proportion of the load which is to be distributed from the processor considered to the other processors involved.

For each processor, there are individual regulators which divide between the other processors the load that is to be distributed.

The method is designed as a "fire-fighting measure" which only takes effect when a processor reaches overload and, if appropriate, tasks (calls) are rejected, rather the load distribution commences earlier and in a softer fashion. As a result, continuous unbalanced load states can be processed better and with fewer rejected tasks (calls).

In the method according to the present invention, the load states which are distributed to the other processors are consistently determined on the basis of the estimated offered load and not on the basis of the actually processed load.

The present method does not require a load balancing flag which regulates entry into the load distribution. The entry is regulated by way of the distribution quota q_i. Furthermore, mutual dependencies between the load states and the load

balancing flag have been eliminated as a result of the absence of a load balancing flag. As a result, the algorithm more easily can be subsequently adapted to changed conditions.

The load-dependent alteration of the individual regulators (load distribution factors p_{ij}) takes place as a function of the number n of processors participating in the load distribution. Consequently, the method is independent of the number of processors involved.

The load-dependent alteration of the distribution quotas and of the individual regulators per control interval takes place in such a way as to avoid excessively slow "creeping" to the optimum value.

The load-dependent alteration of the individual regulators prevents the values from staying at the setting of the preceding load distribution period during a period without load distribution. Rather, there is regulation back to an initial setting.

The inertia, known from the prior art, in the alteration of the quotas has been removed in order to enable easier tracking to the load situation that is actually present.

Attention is supplementary drawn to the definition of a few terms in this application:

The term or word element "quota" describes the fraction of a whole with a range of values between 0 and 1.

The term or word element "state" describes the instantaneous situation or the instantaneous current value of a quantity. Thus, e.g., the load state of a processor is to be understood as the value of the current load of the processor.

The term hysteresis defines the dependence of a current quantity on its earlier values, its history or its time characteristic.

Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

ABSTRACT OF THE DISCLOSURE

A method for load distribution in a real-time multiprocessor system and to a multiprocessor system, each processor carrying a distribution quota which fixes the proportion of the distributable load which is actually to be distributed. The distribution quota is determined anew at time intervals. The only information required by the other processors for each time interval are load value indicators, which depend on an estimated load. Probabilities indicating how load is transferred from one processor to the others during load distribution are additionally carried. Afterward, on the basis of its distribution quota and its load distribution factors, each processor distributes its distributable load to other processors if it distribution quota exceeds a predetermined value.

In the claims:

On page 19, cancel line 1, and substitute the following left-hand justified heading therefor:

We Claim as Our Invention:

Please cancel claims 1-26, without prejudice, and substitute the following claims therefor:

27. A method for load distribution in a multiprocessor system of a communication system, in which tasks that arise can be processed by a plurality of processors MP_i (where $i = 1,2,...,n$) under real-time conditions, the method comprising the following steps that are repeated at time intervals CI:

determining, via each processor MP_I, a respective actual current load Y_I;

estimating, via each processor MP_i and as a function of previously communicated distribution quotas q_i(old) and a typically distributable proportion V of a typical task, a respective offered load A_i, which leads to a multi-value load indication value $MPbi_i$, the distribution quota q_i representing a load proportion which can be distributed to other processors MP_k;

communicating, via each processor MP_I, the respective load indication value $MPbi_i$ to the respective other processors MP_k (where $k = 1,2,...i-1,i+1,...n$);

determining, via each processor MP_I, respective load distribution probabilities p_{ij} (where $j = 1,2,...n$) as a function of the load indication values $MPbi_k$ of the other processors MP_k;

determining, via each processor MP_I, a new distribution quota q_i(new) as a function of the respective current load Y_i and the load distribution factors p_{ij}; and

distributing, via each processing MP_i and based on the respective new distribution quota q_i and the respective load distribution factors p_{ij}, the respective distributable load to the other processors MP_k if the respective new distribution quota q_i(new) exceeds a predetermined value q_v.

28. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein the estimated offered load A_i of the processor MP_i is calculated according to the formula $A_i:=Y_i/(1-q_iV)$.

29. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein the multi-value load indication value $MPbi_i$ can assume three discrete values, NORMAL, HIGH and OVERLOAD, where NORMAL corresponds to a processor capacity utilization of from 0 to 70%, HIGH corresponds to a processor capacity utilization of from 70% to 85% and OVERLOAD corresponds to a processor capacity utilization of from 85% to 100%.

30. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 29, wherein the multi-value load indication value $MPbi_i$ is subject to a hysteresis with regard to changes.

31. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein one of an average and a maximum distributable proportion of a typical task CallP is regarded as the typical distributable proportion V.

32. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 31, wherein the one of the average and the maximum distributable proportion of a typical task is continually determined as moving average and moving maximum value, respectively, over a predetermined time period t_D.

33. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 32, wherein the following holds true for the predetermined time period t_D: $t_D \gg CI$.

34. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 31, wherein one of an average and a maximum task is assumed as the typical task.

35. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 34, wherein the one of the average and the maximum task is continually determined as moving average and moving maximum value, respectively, over a predetermined time period t_D.

36. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 35, wherein the following holds true for the predetermined time period t_D: $t_D \gg CI$.

37. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.05<q_V<0.3$.

38. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein the following holds true for

the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.1<q_V<0.25$.

39. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $q_V = 0.2$.

40. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 29, wherein the calculation of the distribution quota q_i satisfies the following criteria:

p_{ii}: = 0

if $MPbi_j$ corresponds to an average load, preferably $MPbi_j$ = NORMAL, the following holds true:

$p_{ij}(new) = pij(old)+p_{c1}/n$, for j=1,..., and $i \neq j$;

if $MPbi_j$ corresponds to a high load, preferably $MPbi_j$=HIGH, the following holds true:

$p_{ij}(new) = pij(old)-p_{c2}/n$, for j=1,...,n and $i \neq j$;

if $MPbi_j$ corresponds to an overload, preferably $MPbi_j$=OVERLOAD, the following holds true:

$p_{ij}(new) = 0$;

in which case the $p_{ij\ (j=1,\dots,n)}$ are normalized to 1 with the sum p_{sum} of the p_{ij}; and

as initialization value at the beginning of the distribution processes, all p_{ij}, excluding p_{ii}, are identical.

41. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant p_{c1}:

$0.1<p_{c1}<0.5$.

42. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant p_{c1}:

$0.2<p_{c1}<0.3$.

43. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant p_{c1}:

$p_{c1} = 0.25$.

44. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant p_{c2}: $0.1<p_{c2}<0.5$

45. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant p_{c2}: $0.2<p_{c2}<0.3$.

46. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the following holds true for the constant $p_{c2} = 0.25$.

47. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein the initialization value of the p_{ij} at the beginning of the distribution processes is set to be equal to $(n-1)^{-1}$.

48. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 40, wherein each processor MP_i determines a multi-value load status $MPls_i$ based on the respective actual current

load Y_i, and the calculation of the load indication values $MPbi_i$ satisfies the following criteria:

if $MPls_i$ corresponds to the highest load, the following holds true:

$q_i(new)=c_{q1}$;

if $p_{sum} \geq 1$ holds true:

- if the actual current load Y_i is greater than a predetermined value $threshold_H$, q_i is increased where $q_i=\min\{q_i+c_{q1},1\}$,

- if the actual current load Y_i is less than a predetermined value $threshold_N$, q_i is decreased where $q_i=\max\{q_i-c_{q2}, c_{q3}\}$, where $0<c_{q3}<q_v$, preferably $c_{q3}=0.1$,

-if $threshold_N \leq Y_i \leq threshold_H$, qi obtains an intermediate value between the two alternatives mentioned above by linear interpolation; and

if $p_{sum} \leq 1$ holds true: $q_i(new)=q_i(old) * p_{sum}$.

49. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 48, wherein the multi-value load status $MPls_i$ is subject to a hysteresis with regard to changes.

50. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 48, wherein the multi-value load status $MPls_i$ can assume four discrete values, NORMAL (0 to 0.7), HIGH (0.7 to 0.85), OVERLOAD (0.85 to 1) and EXTREME (if load status over a plurality of CI OVERLOAD).

51. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant c_{q1}: $0.05<c_{q1}<0.3$.

52. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $c_{q1}<0.2$.

53. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $c_{q1}=0.15$.

54. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant c_{q2}: $0.05<c_{q2}<0.2$.

55. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $c_{q2}=0.10$.

56. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $threshold_N$: $0.6 < threshold_N < 0.8$.

57. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $threshold_N = 0.7$.

58. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $threshold_H$: $0.7 < threshold_H < 0.95$.

59. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 49, wherein the following holds true for the constant $threshold_H = 0.85$.

60. A method for load distribution in a multiprocessor system of a communication system as claimed in claim 27, wherein an overload value OL_i of the processors MP_i is additionally determined for the purpose of quantifying the overload state of the processors, which overload value is a measure of the magnitude of the overload, where $OL_i=0,1,...m$ and the distribution quota q_i is increased if the magnitude of OL_i becomes greater than 0 and $q_i(new):=\min\{qi(old)+c_{q1},1\}$ is set.

61. A multiprocessor system of a communication system, having a plurality of processors MP_i (where $i=1,2,...,n$) for executing tasks that arise under real-time conditions, each of the plurality of processors comprising:

means for determining a respective actual current load Y_i and for estimating as a function of previously communicated distribution quotas $q_i(old)$ and a typically distributable proportion V of a typical task a respective offered load A_i, which leads to a multi-value load indication value $MPbi_i$, the distribution quota q_i representing a load proportion which can be distributed to other processors MP_k;

means for communicating the respective multi-value load indication value $MPbi_i$ to the other processors MP_k (where $k = 1,2,...i-1,i+1,...n$);

means for determining respective load distribution probabilities p_{ij} (where $j = 1,2,...n$) as a function of the load indication values $MPbi_k$ of the other processors MP_k;

means for determining a new distribution quota $q_i(new)$ as a function of the respective actual current load Y_I; and

means for distributing, on the basis of the quota q_i and the load distribution factors p_{ij}, a respective distributable load to the other processors MP_k if the new distribution quota $q_i(new)$ exceeds a predetermined value q_v.

REMARKS

The present amendment makes editorial changes and corrects typographical errors in the specification, which includes the Abstract, in order to conform the specification to the requirements of United States Patent Practice. No new matter is added thereby. Attached hereto is a marked-up version of the changes made to the specification by the present amendment. The attached page is captioned "**Version With Markings To Show Changes Made".**

In addition, the present amendment cancels original claims 1-26 in favor of new claims 27-61. Claims 27-61 have been presented solely because the revisions by red-lining and underlining which would have been necessary in claims 1-26 in order to present those claims in accordance with preferred United States Patent Practice would have been too extensive, and thus would have been too burdensome. The present amendment is intended for clarification purposes only and not for substantial reasons related to patentability pursuant to 35 USC §§103, 102, 103 or 112. Indeed, the cancellation of claims 1-26 does not constitute an intent on the part of the Applicants to surrender any of the subject matter of claims 1-26.

Early consideration on the merits is respectfully requested.

Respectfully submitted,

(Reg. No. 39,056)
William E. Vaughan
Bell, Boyd & Lloyd LLC
P.O. Box 1135
Chicago, Illinois 60690-1135
(312) 807-4292
Attorneys for Applicants

VERSIONS WITH MARKINGS TO SHOW CHANGES MADE

In The Specification:

The Specification of the present application, including the Abstract, has been amended as follows:

SPECIFICATION

TITLE

~~**Load distribution method of a multiprocessor system, and multiprocessor system**~~

LOAD DISTRIBUTION METHOD OF A MULTIPROCESSOR SYSTEM, AND MULTIPROCESSOR SYSTEM

BACKGROUND OF THE INVENTION

~~Description~~

Field of the Invention

The present invention relates to a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a ~~plurality~~ number of processors MP_i (where i=1,2,...,n) under real-time conditions~~, and~~ . The present invention further relates to a multiprocessor system, in particular of a communication system, having a load distribution mechanism.

Description of the Prior Art

A similar method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, is disclosed, for example, in the applicant's European patent application EP 0 645 702 A1. This document discloses a method for load balancing in a multiprocessor system, in particular a multiprocessor system of a communication system, in which tasks that arise can be processed by a ~~plurality~~ number of processors under real time conditions, in which case, in order to perform the load balancing, ~~generally~~ the following method steps are mentioned:

- each processor determines its load state in the form of a quantified magnitude,

- the load states of the other processors are communicated to each processor within a time frame,

- depending on its load state exceeding a specific magnitude and depending on the load states of the remaining processors, each processor outputs at least a portion of the tasks arising in it to the remaining processors, and

- the output tasks are divided between the remaining processors in accordance with the load states thereof.

In the ~~exemplary embodiment~~ example, the method is concretized to the effect that distribution quotas are calculated during operation continually and before entry into the load distribution~~,~~ which, in this case, does not begin until after a specific overload has been reached, according to which distribution quotas the individual processors output their distributable load to other processors in the case of overload. If the system is permanently utilized to capacity in a nonuniform manner, then the load is distributed only in the event of overload of one or more processors. This is accompanied by unnecessary load rejection, however. Reducing the overload threshold to a lower value does not lead to a satisfactory result because ~~then~~ an unnecessarily large amount of load is distributed and oscillation states can arise. This situation emerges from the assumption made there that the overload or the nonuniform loading lasts for a short duration.

Furthermore, reference is made to Evans D.J. et al., "Dynamic Load Balancing using Task Transfer Probabilities" in Parallel Computing, Vol. 19, No. 8, August 1, 1993, pages 897-916, which also presents a load distribution strategy taking account of load distribution probabilities, although exclusively currently measured values are taken into consideration. The temporal development of the parameters considered is not taken into account.

It is an object of the invention, therefore, to specify an improved load distribution method for a multiprocessor system which commences in good time and in a "soft" fashion and thereby eliminates permanent unbalanced load states in the load offer without load rejection. Moreover, the intention is also to specify a corresponding multiprocessor system.

3/PRTS

09/889890

JC17 Rec'd PCT/PTO 23 JUL 2001

SUMMARY OF THE INVENTION

~~The object is achieved on the one hand by means of a method having the method steps of the first method claim and on the other hand by means of a multiprocessor system having the features of the first apparatus claim.~~

Accordingly, the inventors propose a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a ~~plurality~~ number of processors MP_i (where $i = 1,2,\ldots,n$) under real-time conditions, having the following iterative method steps that are repeated at time intervals CI:

- each processor MP_i determines its actual load state Y_i and estimates as a function of previously communicated distribution quotas $q_i(old)$ (where q_i = load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,

- each processor MP_i indirectly or directly communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where $k = 1,2,\ldots i\text{-}1,i\text{+}1,\ldots n$),

- each processor MP_i determines its load distribution factors p_{ij} (where $j = 1,2,\ldots n$) as a function of the load indication values $MPbi_k$ of ~~said~~ the other processors MP_k,

- each processor MP_i determines its distribution quota $q_i(new)$ as a function of its actual load state Y_i and the load distribution factors P_{ij},

- on the basis of its quota q_i and its load distribution factors p_{ij}, each processor MP_i distributes its distributable load to other processors MP_k if its distribution quota $q_i(new)$ exceeds a predetermined value q_v.

In order to estimate the offered load A_i of a processor MP_i, it is advantageous to use the formula $A_i{:=}Y_i/(1\text{-}q_iV)$. A_i and Y_i can be specified in units of erlangs, while the variables q_i and V are dimensionless fraction indications in accordance with their meaning.

It is also advantageous to subdivide the multi-value load indication value (balancing indicator) $MPbi_i$ into three discrete values, preferably the following

demarcation with threshold values holding true: NORMAL for $MPbi_i$ if the processor capacity utilization is from 0 to 70%, HIGH for $MPbi_i$ if the processor capacity utilization is from 70% to 85%, and OVERLOAD for $MPbi_i$ if the processor capacity utilization is above 85%.

It is also advantageous if a hysteresis is introduced in the case of a load state alteration on account of threshold value overshooting or threshold value undershooting in the case of rising or falling processor capacity utilization.

Moreover, it may be advantageous if the load indication value (balancing indicator) $MPbi_i$ is subject to a temporal hysteresis with regard to changes and thus experiences a certain inertia. Values of 1 to 2 time intervals CI can ~~advantageously~~ be assumed as hysteresis limit.

Further advantageous assumptions in the performance of the method according to the present invention are: the typical distributable proportion V of a typical task shall be the average or maximum proportion, and an average or maximum processing time of a task shall be assumed as the typical processing time of a task. In this case, the respective average value or maximum value of a proportion and respectively of a task ~~can~~ also can ~~advantageously~~ be continually determined during the operating time and, if appropriate, be incorporated as a moving value and be adopted in updated form into the load distribution method. It is favorable here if the time duration over which the moving values are determined is long relative to the control interval CI.

It is also particularly advantageous if the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.05<q_V<0.3$, preferably $0.1<q_v<0.25$, preferably $q_v=0.2$.

Furthermore, the method according to the present invention can be configured particularly advantageously if the following criteria are satisfied in the calculation of the distribution quota q_i:

- $p_{ii}: = 0$

- if $MPbi_j$ corresponds to an average load, preferably $MPbi_j$ = NORMAL, the following holds true:

$p_{ij}(new) = pij(old)+p_{c1}/n$, for $j=1,...,n$ and $i \neq j$

- if $MPbi_j$ corresponds to a high load, preferably $MPbi_j$=HIGH, the following holds true:

$p_{ij}(new) = p_{ij}(old)-p_{c2}/n$, for $j=1,...,n$ and $i \neq j$

- if $MPbi_j$ corresponds to an overload, preferably $MPbi_j$=OVERLOAD, the following holds true:

$p_{ij}(new) = 0$

- in which case the $p_{ij\ (j=1,...,n)}$ is normalized to 1 with the sum p_{sum} of the p_{ij} and

- as initialization value at the beginning of the distribution processes, all p_{ij}, excluding p_{ii}, are identical.

As advantageous numerical values, $0.1<p_{c1}<0.5$, preferably $0.2<p_{c1}<0.3$ and preferably $p_{c1}=0.25$ may be assumed for the constant p_{c1}. Equally, it is advantageous to set $0.1<p_{c2}<0.5$, preferably $0.2<p_{c2}<0.3$, preferably $p_{c2}=0.25$ for the constant p_{c2}. Moreover, the initialization value of the p_{ij} at the beginning of the distribution processes can be set to be equal to $(n-1)^{-1}$.

Furthermore, the method according to the present invention can be configured particularly advantageously if each processor Mp_i determines a multi-value load status (load state) $MP1s_i$ on the basis of its actual current load Y_i, and the following criteria are satisfied in the calculation of the load indication values $MPbi_i$:

- if $MPls_i$ corresponds to the highest load, preferably $MPls_i$=EXTREME, the following holds true:

$q_i(new)=c_{q1}$,

- if $p_{sum} \geq 1$ holds true:

- if the actual load state Y_i is greater than a predetermined value $threshold_H$, q_i is increased where $q_i=\min\{q_i+c_{q1},1\}$,

- if the actual load state Y_i is less than a predetermined value $threshold_N$, q_i is decreased where $q_i=\max\{q_i-c_{q2}, c_{q3}\}$, where $0<c_{q3}<q_v$, preferably $c_{q3}=0.1$,

- otherwise ($threshold_N \leq Y_i \leq threshold_H$), qi obtains an intermediate value between the two alternatives mentioned above, preferably by linear interpolation

- if $p_{sum} \leq 1$ holds true: $q_i(new)=q_i(old) * p_{sum}$.

With regard to the multi-value load status (load state) $MPls_i$, the assumption of four discrete values is proposed as being particularly preferred, the following preferably being assumed: NORMAL for $MPls_i$ if the processor capacity utilization lies below 70%, HIGH for $MPls_i$ if the processor capacity utilization is from 70% to 85%, OVERLAND for $MPls_i$ if the processor capacity utilization lies above 85%, and EXTREME for $MPls_i$ if the load state OVERLAND permanently prevails. In this case, two, is maybe advantageous if the load status (load state) $MPls_i$ is subject to a hysteresis with regard to changes. Values of 1 to 2 time intervals CI ~~can~~ advantageously can be assumed as hysteresis limit.

For optimal configuration of the method of the present invention, the following ranges of numbers of numerical values are preferred for the constant c_{q1}: $0.05<c_{q1}<0.3$, preferably $0.1<c_{q1}<0.2$, preferably $c_{q1}=0.15$. Moreover, preferably $0.05<c_{q2}<0.2$, preferably $c_{q2}=0.10$, can be assumed for the constant c_{q2}.

With regard to the constant $threshold_N$, the following is regarded as a preferred range of values: $0.6< threshold_N <0.8$, preferably $threshold_N=0.7$.

With regard to the constant $threshold_H$, the following is regarded as a preferred range of values: $0.7< threshold_H <0.95$, preferably $threshold_H=0.85$.

Another configuration of the method according to the present invention provides for an overload value OL_i of the processors MP_i to be additionally determined in each time interval CI, which value is a measure of the magnitude of the overload and serves as a benchmark for overload rejection, where $OL_i=0,1,...m$, OL_i representing a quantification for the overload of the processor, and the distribution quota q_i to be increased in any case if $OL_i>0$ where $q_i(new):=\min\{qi(old)+c_{q1},1\}$.

According to the present invention, it is also possible to adapt the load distribution method to changing boundary conditions by the above-specified constants (q_v, p_{c1}, p_{c2}, q_{c1}, q_{c2}, $threshold_N$, $threshold_N$, c_{q1}, c_{q2}, c_{q3}) being at least partly adapted during operation.

The present invention additionally proposes a multiprocessor system, in particular of a communication system, having a ~~plurality~~ number of processors MP_i (where i=1,2,...,n) for executing tasks that arise under real-time conditions, in which case:

- each processor MP_i ~~has means for determining~~ determines its actual load state Y_i, and ~~for estimating~~ estimates as a function of previously communicated distribution quotas q_i(old) (where q_i=load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,

- each processor MP_i ~~has means for~~ indirectly or directly ~~communicating~~ communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where k = 1,2,...i-1,i+1,...n),

- each processor MP_i ~~has means for determining~~ determines its load distribution probabilities p_{ij} (where j = 1,2,...n) as a function of the load indication values $MPbi_k$ of ~~said~~ the other processors MP_k,

- each processor MP_i ~~has means for determining~~ determines its distribution quota q_i(new) as a function of its actual load state Y_i and

- each processor MP_i ~~has means for distributing~~ distributes, on the basis of its quota q_i and its load distribution factors p_{ij},

its distributable load to other processors MP_k if its distribution quota q_i(new) exceeds a predetermined value q_v.

According to the present invention, the multiprocessor system proposed above can be configured such that one of the abovementioned methods is implemented in each case, the implementation being effected by corresponding programming of the processors.

It should also be pointed out that the index (old) relates, in each case, to the values of the preceding iteration step, and the index (new) relates to the now current iteration step.

The particular advantage of the method according to the present invention and of the corresponding multiprocessor system is that, in contrast to the prior art mentioned in the introduction, it ensures a "soft" entry into the load distribution and, as a result, is more adaptable and less susceptible to unbalanced load situations~~, and~~ wherein oscillation states are better avoided ~~better~~. Ultimately, this reduces the probability of the rejection of tasks, in particular switching tasks.

~~Further configurations, additional features and advantages of the invention emerge from the following description of a preferred exemplary embodiment with reference to the drawings.~~

~~It is understood that the features of the invention that have been mentioned above and will be explained below can be used not only in the combination respectively specified but also in other combinations or by themselves, without departing from the scope of the invention.~~

Additional features and advantages of the present invention are described in, and will be apparent from, the following Detailed Description of the Preferred Embodiments and the Drawings.

DESCRIPTION OF THE DRAWINGS

~~Specifically, in the figures:~~

Figure 1~~:~~ shows a flow diagram of ~~the~~ an arising and distributed load offer;

Figure 2a~~:~~ shows a graphical illustration of the decisions for updating the load distribution factors p_{ij};

Figure 2b~~:~~ shows a graphical illustration of the decisions for updating the distribution quotas q_I; and

Figure 3~~:~~ shows a formula for linear interpolation of q_I.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The method according to the present invention (normal load balancing=NLB) is a load balancing method in which quotas are set and which runs

on a multiprocessor system, in particular in a switching center of a communication system, for distributing operating loads that arise between the respective other processors~~, and~~ It is intended to ensure that lengthy unbalanced load situations are eliminated, and as far as possible, all requested tasks are processed in the shortest possible time. A particularly advantageous embodiment of this method will be described below.

Each processor MP_i where i=1,2,...,n carries a distribution quota q_i, which fixes the proportion V of the distributable load which is actually to be distributed. Such a quota enables a softer entry or exit from the load distribution to other processors. Oscillation states and load fluctuations are avoided in this way. This may be the case, for example, if a processor distributes so much load to another processor that the latter is in turn overloaded.

The distribution quota q_i is determined anew at each time interval CI. The only information required by the other processors MP_k where k=1,...i-1,i+1,...n for each CI are load value indicators (balancing indicators) $MPbi_i$. These load value indicators are, similarly to the load status values (load states)from the load control, load states having the significances NORMAL, HIGH or OVERLOAD. While the load state is determined on the basis of the actually processed load Y_i of the processor MP_i, the load value indicator $MPbi_i$ is determined from an estimation of the currently offered load A_i. The estimated offered load A_i may, due to load distribution, be considerably more than the actually processed load Y_i and constitutes the crucial quantity which (in the form of the load value indicator $MPbi_i$) is made available as information by one processor MP_i to the others MP_k.

In addition to the distribution quota q_i, each MP_i carries probabilities p_{ij} which indicate the probability that, in the event of load distribution, load will be transferred from the i-th processor MP_i to the j-th processor MP_j. The probabilities are determined in such a way that if, for instance, the j-th processor MP_j already has a large amount of load to process and, therefore, can only take up a small amount of additional load, the associated p_{ij} is less than the P_{ik} for a free MP_k.

Figure 1 illustrates the interaction of the p_{ij} and q_i. The double indexing "ij" of the characteristic quantities ~~means that~~ refers to the respective processor with the number of the first index (here i) in each case ~~knows~~ knowing a "column" of n values with the second index (here j). It should be noted that each processor only knows its relevant values (that is to say its column), overall a square matrix being known in the system. Thus, for example, p_{ij} is the probability that load will be distributed from the i-th MP to the j-th MP if the i-th MP has too much load.

In Figure 1, moreover, the actually processed load of the j-th processor MP_j is designated by Y_j, the estimated offered load is designated by A_j and that part of the load offer which can be shifted is designated by a. The load situation shown is overload (OVERLOAD) on MP_1, there still being space for additional tasks on the MP_k where k=2,3,4. ~~The figure~~ Figure 1 shows how the MP_1 processes a first part of the load itself and distributes the remainder a. Of this remainder a, the largest proportion goes to MP_3 and the smallest proportion to MP_4, which, in this example, ~~thus~~ already has a large amount of its own load to process. The loads which the MP_k additionally receive besides that from MP_1 are not depicted. The width of the flow bars represents a measure of the magnitude of the load.

The following algorithm is thus produced in accordance with the concept of the present invention: if the j-th processor MP_j reports the balancing indicator NORMAL, the p_{ij} is increased on the MP_i respectively considered. The probability that this processor MP_i will output load to MP_j if it has to distribute load thus rises. If the balancing indicator HIGH is reported, then the p_{ij} is decreased. If the balancing indicator OVERLOAD is reported, p_{ij} is set to zero, with the result that no load is output to the j-th processor MP_j. The distribution quota q_i is changed following the determination of the p_{ij}. If many of the p_{ij} were able to be increased, then the sum of the p_{ij} over j is greater than 1 and there is evidently still space on the other processors MP_k. The distribution quota q_i can, thus, be changed according to the requirements of the processor (considered).

The distribution quota q_i is increased in the event of high load Y_i on the processor MP_i considered, and q_i is decreased in the event of low load. If many of the p_{ij} have

been reduced, then the sum of the p_{ij} over j is less than 1 and the distribution quota q_i must be reduced.

An illustration of these decisions is represented in ~~figures~~ Figures 2 and 2b. The decision diagrams show the updating algorithms for p_{ij} (Figure 2a) and for distribution quota q_i (Figure 2b), which are carried out in each time interval CI for the i-th processor MP_i.

In the load distribution method (NLB) according to the present invention, some parameters (constants) are required, the choice of which can greatly influence the behavior in specific load situations. In most cases this results in a conflict between a load distribution method which can react rapidly to load changes, and a stable load distribution method which does not tend toward oscillations and further distribution of tasks. In this case, "further distribution" ~~means~~ refers to the simultaneous distribution of dedicated load and the processing of extrinsic load on one processor.

The following parameter changes bring about a more rapidly reacting NLB:

- the relatively great alteration of q_i where: $0.15<c_{q1}$, $0.1<c_{q2}$
- the relatively great alteration of the p_{ij} where: $0.25<p_{c1}$, $0.25<p_{c2}$
- the relatively late setting of the load indication values $MPbi_i$ where: $threshold_H>0.7$ (i.e. report only in the event of relatively high load 'HIGH') to the other processors MPk.

In detail, the preferred method thus proceeds as follows in a multiprocessor communication computer:

As the duration of the time interval (control interval) CI of the time frame with which the method iteratively proceeds, 1 to 2 seconds is preferably chosen in the case of the presently known multiprocessor systems appertaining to switching technology. It goes without saying that the time interval can be shortened with rising processor power.

The quantities q_i, p_{ij}, $MPls_i$ and MPb_{ij} are updated in each control interval CI.

The actually processed load Y_i of a processor MP_i is determined as processor run time quantity, measured in erlangs.

The estimated offered load A_j of a processor MP_i is determined from the distribution quota q_i of the current control interval CI and the estimated distributable proportion of an average task, for example the processing of a call.

The following holds true:

The number of processors MP_i in the multiprocessor system is n.

$A_i := Y_i/(1-q_iV)$, where V is the distributable proportion of a call.

$MPls_i$: load state of the i-th MP, can assume the values NORMAL, HIGH, OVERLOAD or EXTREME. The actually processed load Yi is used to calculate the load state.

In order to avoid premature changes of the $MPls_i$, hystereses are introduced. If, for instance, the $MPls_i$ is set from NORMAL to HIGH, it must be the case that $Y_i > threshold_N + \Delta_+$, whereas, in order to get from HIGH to NORMAL, it must be the case that $Y_i < threshold_N - \Delta_-$. This procedure is also known as the high water-low water method. In the case of EXTREME, the distribution method (load balancing) must be switched off for this processor MP_i, for system engineering reasons relating to the switching center.

$threshold_N$: is the normal load threshold - after taking a hysteresis into account, the MPls is recorded as NORMAL below the said threshold and as HIGH above said threshold.

$threshold_H$: High load threshold - after taking a hysteresis and a load-dependent temporal delay (start indicator) into account, the MPls is recorded as HIGH below this threshold and as OVERLOAD above said threshold.

The load indication value (balancing indicator) $MPbi_i$ of the i-th processor MP_i can assume the values NORMAL, HIGH or OVERLOAD. This value is calculated like the $MPls_i$, except that here, instead of the actual load Y_i, the estimated offered load A_i is taken as a basis and other values are adopted for Δ_+ and Δ_-, where $\Delta_+ = \Delta_- = 0.02$.

In addition, an Overload Level OL_i of the processor MP_i is determined, which can assume the values 0... 6 and is conceived as quantification of the overload state of the processor MP_i. If the $OL_i>0$, calls are rejected; the higher the value, the greater the probability that a call will be rejected.

The load which is to be distributed from MP_i to MP_j is expressed as a probability p_{ij} and can, thus, assume values between 0 and 1.

The magnitude of the value p_{ij} is determined by the following criteria:

- initialize p_{ij} where $p_{ij}:=(n-1)^{-1}$

- p_{ii}: 0, MP_i should not distribute to itself.

- If $MPbi_j$=NORMAL: $p_{ij} \rightarrow p_{ij} + 0.25/n$, $j=1,...,n$, $i \neq j$. The old p_{ij} can be increased because there is still space on the processor MP_j.

- If $MPbi_j$=HIGH: $p_{ij} \rightarrow p_{ij} - 0.25/n$. The old p_{ij} must be decreased because MP_j is utilized to full capacity.

- If $MPbi_j$=OVERLOAD: $p_{ij} = 0$. No load should be output to overloaded processors MP_n.

The newly determined p_{ij} must still be normalized:

Set p_{sum} = sum (p_{ij}) over $j=1,...,n$ and normalize (if $p_{sum} > 0$) where

$p_{ij} \rightarrow p_{ij}/p_{sum}$

Afterward, the distribution quota q_i is determined using the following criteria:

- Initialization value: $q_i = 0.1$

- If the $MPls_i$ =EXTREME: $q_i = 0.1$. This MP is overloaded so severely that even its own proportion for a distributed call would overtax it. Therefore, no load balancing, rather only rejecting; load balancing is not practical, moreover, for system engineering reasons relating to the switching center.

- If $p_{sum} > 1$, more load can evidently be distributed. q_i can then be determined according to the requirements of the MP_i, where:

1. If the $OL_i > 0$, increase q_i in any case, where: $q_i \rightarrow \min \{q_i + 0.15, 1\}$

2. If $Y_i > threshold_H$, increase q_i, where: $q_i \rightarrow \min \{q_i + 0.15, 1\}$

3. If $Y_i < threshold_N$, decrease q_i, where: $q_i \rightarrow \max \{q_i - 0.10, 0.1\}$

4. Otherwise, if $threshold_N < Y_i < threshold_H$ the following holds true:

$q_i \rightarrow \min\{\max\{q_i+(0.25/threshold_H-threshold_N)) * (Y_i-threshold_N)-0.1, 0.1\}, 1.0\}$.

This is the linear interpolation between the above increase by 0.15 and the above decrease by 0.1. The formula is represented again more readably in ~~figure~~ Figure 3.

- If $p_{sum} < 1$, evidently too much load was distributed and q_i must be decreased, where: $q_i \rightarrow q_i * p_{sum}$.

- The processor MP_i distributes load to other processors MP_k if it becomes the case that $q_i > 0.25$.

The method according to the present invention thus has the following properties and advantages:

A very small information overhead exists between the processors participating in the load distribution method. Only a few, preferably three-value, load states are reciprocally known, which load states are updated and distributed only once per control interval.

For each processor, there is a quota which is updated in each control interval and regulates the proportion of the load which is to be distributed from the processor considered to the other processors involved.

For each processor, there are individual regulators which divide between the other processors the load that is to be distributed.

The method is ~~not only~~ designed as a "fire-fighting measure" which only takes effect when a processor reaches overload and, if appropriate, tasks (calls) are rejected, rather the load distribution commences earlier and in a softer fashion. As a result, continuous unbalanced load states can be processed better and with fewer rejected tasks (calls).

In the method according to the present invention, the load states which are distributed to the other processors are consistently determined on the basis of the estimated offered load and not on the basis of the actually processed load.

The present method does not require a load balancing flag which regulates entry into the load distribution. The entry is regulated by way of the distribution quota q_i. Furthermore, mutual dependencies between the load states and the load balancing flag have been eliminated as a result of the absence of a load balancing flag. As a result, the algorithm ~~can~~ more easily can be subsequently adapted to changed conditions.

The load-dependent alteration of the individual regulators (load distribution factors p_{ij}) takes place as a function of the number n of processors participating in the load distribution. Consequently, the method is independent of the number of processors involved.

The load-dependent alteration of the distribution quotas and of the individual regulators per control interval takes place in such a way as to avoid excessively slow "creeping" to the optimum value.

The load-dependent alteration of the individual regulators prevents the values from staying at the setting of the preceding load distribution period during a period without load distribution. Rather, there is regulation back to an initial setting.

The inertia, known from the prior art, in the alteration of the quotas has been removed in order to enable easier tracking to the load situation that is actually present.

Attention is supplementary drawn to the definition of a few terms in this application:

The term or word element "quota" describes the fraction of a whole with a range of values between 0 and 1.

The term ~~of~~ or word element "state" describes the instantaneous situation or the instantaneous current value of a quantity. Thus, e.g., the load state of a processor is to be understood as the value of the current load of the processor.

The term hysteresis defines the dependence of a current quantity on its earlier values, its history or its time characteristic.

Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

~~Abstract~~

ABSTRACT OF THE DISCLOSURE

~~Load distribution method of a multiprocessor system, and multiprocessor system~~

~~The invention relates to a~~ A method for load distribution in a real-time multiprocessor system and to a multiprocessor system, each processor carrying a distribution quota which fixes the proportion of the distributable load which is actually to be distributed. The distribution quota is determined anew at time intervals. The only information required by the other processors for each time interval are load value indicators, which depend on an estimated load. Probabilities indicating how load is transferred from one processor to the others during load distribution are additionally carried. Afterward, on the basis of its distribution quota and its load distribution factors, each processor distributes its distributable load to other processors if it distribution quota exceeds a predetermined value.

~~Figure 1~~

Description

Load distribution method of a multiprocessor system, and multiprocessor system

The invention relates to a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a plurality of processors MP_i (where $i=1,2,...,n$) under real-time conditions, and to a multiprocessor system, in particular of a communication system, having a load distribution mechanism.

A similar method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, is disclosed for example in the applicant's European patent application EP 0 645 702 A1. This document discloses a method for load balancing in a multiprocessor system, in particular a multiprocessor system of a communication system, in which tasks that arise can be processed by a plurality of processors under real time conditions, in which case, in order to perform the load balancing, generally the following method steps are mentioned:

- each processor determines its load state in the form of a quantified magnitude,
- the load states of the other processors are communicated to each processor within a time frame,
- depending on its load state exceeding a specific magnitude and depending on the load states of the remaining processors, each processor outputs at least a portion of the tasks arising in it to the remaining processors, and

- the output tasks are divided between the remaining processors in accordance with the load states thereof.

In the exemplary embodiment, the method is concretized to the effect that distribution quotas are calculated during operation continually and before entry into the load distribution, which in this case does not begin until after a specific overload has been reached, according to which distribution quotas the individual processors output their distributable load to other processors in the case of overload. If the system is permanently utilized to capacity in a nonuniform manner, then the load is distributed only in the event of overload of one or more processors. This is accompanied by unnecessary load rejection, however. Reducing the overload threshold to a lower value does not lead to a satisfactory result because then an unnecessarily large amount of load is distributed and oscillation states can arise. This situation emerges from the assumption made there that the overload or the nonuniform loading lasts for a short duration.

Furthermore, reference is made to Evans D.J. et al., "Dynamic Load Balancing using Task Transfer Probabilities" in Parallel Computing, Vol. 19, No. 8, August 1, 1993, pages 897-916, which also presents a load distribution strategy taking account of load distribution probabilities, although exclusively currently measured values are taken into consideration. The temporal development of the parameters considered is not taken into account.

It is an object of the invention, therefore, to specify an improved load distribution method for a multiprocessor system which commences in good time and in a "soft" fashion and thereby eliminates permanent unbalanced load states in the load offer without load rejection. Moreover, the intention is also to specify a corresponding multiprocessor system.

The object is achieved on the one hand by means of a method having the method steps of the first method claim and on the other hand by means of a multiprocessor system having the features of the first apparatus claim.

Accordingly, the inventors propose a method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a plurality of processors MP_i (where $i = 1,2,...,n$) under real-time conditions, having the following iterative method steps that are repeated at time intervals CI:

- each processor MP_i determines its actual load state Y_i and estimates as a function of previously communicated distribution quotas q_i(old) (where q_i = load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,
- each processor MP_i indirectly or directly communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where $k = 1,2,...i-1,i+1,...n$),
- each processor MP_i determines its load distribution factors p_{ij} (where $j = 1,2,...n$) as a function of the load indication values $MPbi_k$ of said other processors MP_k,
- each processor MP_i determines its distribution quota q_i(new) as a function of its actual load state Y_i and the load distribution factors P_{ij},
- on the basis of its quota q_i and its load distribution factors p_{ij}, each processor MP_i distributes its distributable load to other processors MP_k if its

distribution quota $q_i(new)$ exceeds a predetermined value q_v.

In order to estimate the offered load A_i of a processor MP_i, it is advantageous to use the formula $A_i:=Y_i/(1-q_iV)$. A_i and Y_i can be specified in units of erlangs, while the variables q_i and V are dimensionless fraction indications in accordance with their meaning.

It is also advantageous to subdivide the multi-value load indication value (balancing indicator) $MPbi_i$ into three discrete values, preferably the following demarcation with threshold values holding true: NORMAL for $MPbi_i$ if the processor capacity utilization is from 0 to 70%, HIGH for $MPbi_i$ if the processor capacity utilization is from 70% to 85%, and OVERLOAD for $MPbi_i$ if the processor capacity utilization is above 85%.

It is also advantageous if a hysteresis is introduced in the case of a load state alteration on account of threshold value overshooting or threshold value undershooting in the case of rising or falling processor capacity utilization.

Moreover, it may be advantageous if the load indication value (balancing indicator) $MPbi_i$ is subject to a temporal hysteresis with regard to changes and thus experiences a certain inertia. Values of 1 to 2 time intervals CI can advantageously be assumed as hysteresis limit.

Further advantageous assumptions in the performance of the method according to the invention are: the typical distributable proportion V of a typical task shall be the average

or maximum proportion, and an average or maximum processing time of a task shall be assumed as the typical processing time of a task. In this case, the respective average value or maximum value of a proportion and respectively of a task can also advantageously be continually determined during the operating time and, if appropriate, be incorporated as a moving value and be adopted in updated form into the load distribution method. It is favorable here if the time duration over which the moving values are determined is long relative to the control interval CI.

It is also particularly advantageous if the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.05<q_v<0.3$, preferably $0.1<q_v<0.25$, preferably $q_v=0.2$.

Furthermore, the method according to the invention can be configured particularly advantageously if the following criteria are satisfied in the calculation of the distribution quota q_i:

- $p_{ii} := 0$
- if $MPbi_j$ corresponds to an average load, preferably $MPbi_j$ = NORMAL, the following holds true:
  $p_{ij}(new) = pij(old)+p_{c1}/n$, for $j=1,\ldots,n$ and $i\neq j$
- if $MPbi_j$ corresponds to a high load, preferably $MPbi_j$=HIGH, the following holds true:
  $p_{ij}(new) = p_{ij}(old)-p_{c2}/n$, for $j=1,\ldots,n$ and $i\neq j$
- if $MPbi_j$ corresponds to an overload, preferably $MPbi_j$=OVERLOAD, the following holds true:
  $p_{ij}(new) = 0$
- in which case the $p_{ij\ (j=1,\ldots,n)}$ is normalized to 1 with the sum p_{sum} of the p_{ij} and
- as initialization value at the beginning of the distribution processes, all p_{ij}, excluding p_{ii}, are identical.

As advantageous numerical values, $0.1<p_{c1}<0.5$, preferably $0.2<p_{c1}<0.3$ and preferably $p_{c1}=0.25$ may be assumed for the constant p_{c1}. Equally, it is advantageous to set $0.1<p_{c2}<0.5$, preferably $0.2<p_{c2}<0.3$, preferably $p_{c2}=0.25$ for the constant p_{c2}. Moreover, the initialization value of the p_{ij} at the beginning of the distribution processes can be set to be equal to $(n-1)^{-1}$.

Furthermore, the method according to the invention can be configured particularly advantageously if each processor Mp_i determines a multi-value load status (load state) $MP1s_i$ on the basis of its actual current load Y_i, and the following criteria are satisfied in the calculation of the load indication values $MPbi_i$:

- if $MP1s_i$ corresponds to the highest load, preferably $MP1s_i$=EXTREME, the following holds true:

$q_i(new)=c_{q1}$,

- if $p_{sum} \geq 1$ holds true:

- if the actual load state Y_i is greater than a predetermined value $threshold_H$, q_i is increased where $q_i=\min\{q_i+c_{q1},1\}$,

- if the actual load state Y_i is less than a predetermined value $threshold_N$, q_i is decreased where $q_i=\max\{q_i-c_{q2}, c_{q3}\}$, where $0<c_{q3}<q_v$, preferably $c_{q3}=0.1$,

- otherwise ($threshold_N \leq Y_i \leq threshold_H$), qi obtains an intermediate value between the two alternatives mentioned above, preferably by linear interpolation

- if $p_{sum} \leq 1$ holds true: $q_i(new)=q_i(old) * p_{sum}$.

With regard to the multi-value load status (load state) $MP1s_i$, the assumption of four discrete values is proposed as being particularly preferred, the following preferably being assumed: NORMAL for

$MPls_i$ if the processor capacity utilization lies below 70%, HIGH for $MPls_i$ if the processor capacity utilization is from 70% to 85%, OVERLAND for $MPls_i$ if the processor capacity utilization lies above 85%, and EXTREME for $MPls_i$ if the load state OVERLAND permanently prevails. In this case, two, is maybe advantageous if the load status (load state) $MPls_i$ is subject to a hysteresis with regard to changes. Values of 1 to 2 time intervals CI can advantageously be assumed as hysteresis limit.

For optimal configuration of the method, the following ranges of numbers of numerical values are preferred for the constant c_{q1}:

$0.05<c_{q1}<0.3$, preferably $0.1<c_{q1}<0.2$, preferably $c_{q1}=0.15$. Moreover, preferably $0.05<c_{q2}<0.2$, preferably $c_{q2}=0.10$ can be assumed for the constant c_{q2}.

With regard to the constant $threshold_N$, the following is regarded as a preferred range of values: $0.6< threshold_N <0.8$, preferably $threshold_N=0.7$.

With regard to the constant $threshold_H$, the following is regarded as a preferred range of values: $0.7< threshold_H <0.95$, preferably $threshold_H=0.85$.

Another configuration of the method according to the invention provides for an overload value OL_i of the processors MP_i to be additionally determined in each time interval CI, which value is a measure of the magnitude of the overload and serves as a benchmark for overload rejection, where $OL_i=0,1,...m$, OL_i representing a quantification for the overload of the processor, and the distribution quota q_i to be increased in any case if $OL_i>0$ where $q_i(new):=\min\{qi(old)+c_{q1},1\}$.

According to the invention, it is also possible to adapt the load distribution method to changing boundary conditions by the above-specified constants (q_V, p_{c1}, p_{c2}, q_{c1}, q_{c2}, $threshold_N$, $threshold_N$, c_{q1}, c_{q2}, c_{q3}) being at least partly adapted during operation.

The invention additionally proposes a multiprocessor system, in particular of a communication system, having a plurality of processors MP_i (where i=1,2,...,n) for executing tasks that arise under real-time conditions, in which case:

- each processor MP_i has means for determining its actual load state Y_i, and for estimating as a function of previously communicated distribution quotas q_i(old) (where q_i=load proportion to be distributed, if possible, to other processors MP_k) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$,

- each processor MP_i has means for indirectly or directly communicating its load indication value $MPbi_i$ to the respective other processors MP_k (where k = 1,2,...i-1,i+1,...n),

- each processor MP_i has means for determining its load distribution probabilities p_{ij} (where j = 1,2,...n) as a function of the load indication values $MPbi_k$ of said other processors MP_k,

- each processor MP_i has means for determining its distribution quota q_i(new) as a function of its actual load state Y_i and

- each processor MP_i has means for distributing, on the basis of its quota q_i and its load distribution factors p_{ij},

its distributable load to other processors MP_k if its distribution quota q_i(new) exceeds a predetermined value q_v.

According to the invention, the multiprocessor system proposed above can be configured such that one of the abovementioned methods is implemented in each case, the implementation being effected by corresponding programming of the processors.

It should also be pointed out that the index (old) relates in each case to the values of the preceding iteration step, and the index (new) relates to the now current iteration step.

The particular advantage of the method according to the invention and of the corresponding multiprocessor system is that, in contrast to the prior art mentioned in the introduction, it ensures a "soft" entry into the load distribution and, as a result, is more adaptable and less susceptible to unbalanced load situations, and oscillation states are avoided better. Ultimately, this reduces the probability of the rejection of tasks, in particular switching tasks.

Further configurations, additional features and advantages of the invention emerge from the following description of a preferred exemplary embodiment with reference to the drawings.

It is understood that the features of the invention that have been mentioned above and will be explained below can be used not only in the combination respectively specified but also in other combinations or by themselves, without departing from the scope of the invention.

Specifically, in the figures:

Figure 1: shows a flow diagram of the arising and distributed load offer

Figure 2a: shows a graphical illustration of the decisions for updating the load distribution factors p_{ij}

Figure 2b: shows a graphical illustration of the decisions for updating the distribution quotas q_i

Figure 3: shows a formula for linear interpolation of q_i.

The method according to the invention (normal load balancing=NLB) is a load balancing method in which quotas are set and which runs on a multiprocessor system, in particular in a switching center of a communication system, for distributing operating loads that arise between the respective other processors, and is intended to ensure that lengthy unbalanced load situations are eliminated and as far as possible all requested tasks are processed in the shortest possible time. A particularly advantageous embodiment of this method will be described below.

Each processor MP_i where $i=1,2,...,n$ carries a distribution quota q_i, which fixes the proportion V of the distributable load which is actually to be distributed. Such a quota enables a softer entry or exit from the load distribution to other processors. Oscillation states and load fluctuations are avoided in this way. This may be the case, for example, if a processor distributes so much load to another processor that the latter is in turn overloaded.

The distribution quota q_i is determined anew at each time interval CI. The only information required by the other processors MP_k where $k=1,...i-1,i+1,...n$ for each CI are load value indicators (balancing indicators) $MPbi_i$. These load value indicators are - similarly to the load status values (load states)

from the load control - load states having the significances NORMAL, HIGH or OVERLOAD. While the load state is determined on the basis of the actually processed load Y_i of the processor MP_i, the load value indicator $MPbi_i$ is determined from an estimation of the currently offered load A_i. The estimated offered load A_i may, due to load distribution, be considerably more than the actually processed load Y_i and constitutes the crucial quantity which (in the form of the load value indicator $MPbi_i$) is made available as information by one processor MP_i to the others MP_k.

In addition to the distribution quota q_i, each MP_i carries probabilities p_{ij} which indicate the probability that, in the event of load distribution, load will be transferred from the i-th processor MP_i to the j-th processor MP_j. The probabilities are determined in such a way that if, for instance, the j-th processor MP_j already has a large amount of load to process and, therefore, can only take up a small amount of additional load, the associated p_{ij} is less than the p_{ik} for a free MP_k.

Figure 1 illustrates the interaction of the p_{ij} and q_i. The double indexing "ij" of the characteristic quantities means that the respective processor with the number of the first index (here i) in each case knows a "column" of n values with the second index (here j). It should be noted that each processor only knows its relevant values (that is to say its column), overall a square matrix being known in the system. Thus, for example, p_{ij} is the probability that load will be distributed from the i-th MP to the j-th MP if the i-th MP has too much load.

In Figure 1, moreover, the actually processed load of the j-th processor MP_j is designated by Y_j, the estimated offered load is designated by A_j and that part of the load offer which can be shifted is designated by a. The load situation shown is overload

(OVERLOAD) on MP_1, there still being space for additional tasks on the MP_k where k=2,3,4. The figure shows how the MP_1

processes a first part of the load itself and distributes the remainder a. Of this remainder a, the largest proportion goes to MP_3 and the smallest proportion to MP_4, which, in this example, thus already has a large amount of its own load to process. The loads which the MP_k additionally receive besides that from MP_1 are not depicted. The width of the flow bars represents a measure of the magnitude of the load.

The following algorithm is thus produced in accordance with the concept of the invention: if the j-th processor MP_j reports the balancing indicator NORMAL, the p_{ij} is increased on the MP_i respectively considered. The probability that this processor MP_i will output load to MP_j if it has to distribute load thus rises. If the balancing indicator HIGH is reported, then the p_{ij} is decreased. If the balancing indicator OVERLOAD is reported, p_{ij} is set to zero, with the result that no load is output to the j-th processor MP_j. The distribution quota q_i is changed following the determination of the p_{ij}. If many of the p_{ij} were able to be increased, then the sum of the p_{ij} over j is greater than 1 and there is evidently still space on the other processors MP_k. The distribution quota q_i can thus be changed according to the requirements of the processor (considered).

The distribution quota q_i is increased in the event of high load Y_i on the processor MP_i considered, and q_i is decreased in the event of low load. If many of the p_{ij} have been reduced, then the sum of the p_{ij} over j is less than 1 and the distribution quota q_i must be reduced.

An illustration of these decisions is represented in figures 2 and 2b. The decision diagrams show the updating algorithms for p_{ij} (Figure 2a) and for distribution quota q_i (Figure 2b), which are carried out in each time interval CI for the i-th processor MP_i.

In the load distribution method (NLB) according to the invention, some parameters (constants) are required, the choice of which

can greatly influence the behavior in specific load situations. In most cases this results in a conflict between a load distribution method which can react rapidly to load changes, and a stable load distribution method which does not tend toward oscillations and further distribution of tasks. In this case, "further distribution" means the simultaneous distribution of dedicated load and the processing of extrinsic load on one processor.

The following parameter changes bring about a more rapidly reacting NLB:

- the relatively great alteration of q_i where: $0.15<c_{q1}$, $0.1<c_{q2}$
- the relatively great alteration of the p_{ij} where: $0.25<p_{c1}$, $0.25<p_{c2}$
- the relatively late setting of the load indication values $MPbi_i$ where: $threshold_H>0.7$ (i.e. report only in the event of relatively high load 'HIGH') to the other processors MPk

In detail, the preferred method thus proceeds as follows in a multiprocessor communication computer:

As the duration of the time interval (control interval) CI of the time frame with which the method iteratively proceeds, 1 to 2 seconds is preferably chosen in the case of the presently known multiprocessor systems appertaining to switching technology. It goes without saying that the time interval can be shortened with rising processor power.

The quantities q_i, p_{ij}, $MPls_i$ and MPb_{ij} are updated in each control interval CI.

The actually processed load Y_i of a processor MP_i is determined as processor run time quantity, measured in erlangs.

The estimated offered load A_j of a processor MP_i is determined from the distribution quota q_i of the current control interval CI and the estimated distributable proportion of an

average task, for example the processing of a call.

The following holds true:

The number of processors MP_i in the multiprocessor system is n.

$A_i := Y_i/(1-q_iV)$, where V is the distributable proportion of a call.

$MPls_i$: load state of the i-th MP, can assume the values NORMAL, HIGH, OVERLOAD or EXTREME. The actually processed load Yi is used to calculate the load state.

In order to avoid premature changes of the $MPls_i$, hystereses are introduced. If, for instance, the $MPls_i$ is set from NORMAL to HIGH, it must be the case that $Y_i > threshold_N + \Delta_+$, whereas, in order to get from HIGH to NORMAL, it must be the case that $Y_i < threshold_N - \Delta_-$. This procedure is also known as the high water-low water method. In the case of EXTREME, the distribution method (load balancing) must be switched off for this processor MP_i, for system engineering reasons relating to the switching center.

$threshold_N$: is the normal load threshold - after taking a hysteresis into account, the MPls is recorded as NORMAL below the said threshold and as HIGH above said threshold.

$threshold_H$: High load threshold - after taking a hysteresis and a load-dependent temporal delay (start indicator) into account, the MPls is recorded as HIGH below this threshold and as OVERLOAD above said threshold.

The load indication value (balancing indicator) $MPbi_i$ of the i-th processor MP_i can assume the values NORMAL, HIGH or OVERLOAD. This value is calculated like the $MPls_i$, except that here, instead of the actual load Y_i, the estimated offered load A_i is taken as a basis and other values are adopted for Δ_+ and Δ_-, where $\Delta_+ = \Delta_- = 0.02$.

In addition, an Overload Level OL_i of the processor MP_i is determined, which can assume the values 0... 6 and is conceived as quantification of the overload state of the processor MP_i. If the $OL_i>0$, calls are rejected; the higher the value, the greater the probability that a call will be rejected.

The load which is to be distributed from MP_i to MP_j is expressed as a probability p_{ij} and can thus assume values between 0 and 1.

The magnitude of the value p_{ij} is determined by the following criteria:

- initialize p_{ij} where $p_{ij}:=(n-1)^{-1}$
- p_{ii}: 0, MP_i should not distribute to itself.
- If $MPbi_j$=NORMAL: $p_{ij} \rightarrow p_{ij} + 0.25/n$, $j=1,\ldots,n$, $i\neq j$. The old p_{ij} can be increased because there is still space on the processor MP_j.
- If $MPbi_j$=HIGH: $p_{ij} \rightarrow p_{ij} - 0.25/n$. The old p_{ij} must be decreased because MP_j is utilized to full capacity.
- If $MPbi_j$=OVERLOAD: $p_{ij} = 0$. No load should be output to overloaded processors MP_n.

The newly determined p_{ij} must still be normalized:

Set p_{sum} = sum (p_{ij}) over $j=1,\ldots,n$ and normalize (if $p_{sum} > 0$) where $p_{ij} \rightarrow p_{ij}/p_{sum}$

Afterward, the distribution quota q_i is determined using the following criteria:

- Initialization value: $q_i = 0.1$
- If the $MPls_i$ =EXTREME: $q_i = 0.1$. This MP is overloaded so severely that even its own proportion for a distributed call would overtax it. Therefore, no load balancing, rather only rejecting; load balancing is not practical, moreover, for system engineering reasons relating to the switching center.

- If $p_{sum} > 1$, more load can evidently be distributed. q_i can then be determined according to the requirements of the MP_i, where:

1. If the $OL_i > 0$, increase q_i in any case, where: $q_i \rightarrow \min\{q_i + 0.15, 1\}$

2. If $Y_i > threshold_H$, increase q_i, where: $q_i \rightarrow \min\{q_i + 0.15, 1\}$

3. If $Y_i < threshold_N$, decrease q_i, where: $q_i \rightarrow \max\{q_i - 0.10, 0.1\}$

4. Otherwise, if $threshold_N < Y_i < threshold_H$ the following holds true:

$$q_i \rightarrow \min\{\max\{q_i+(0.25/threshold_H-threshold_N)) * (Y_i-threshold_N)-0.1, 0.1\}, 1.0\}$$

This is the linear interpolation between the above increase by 0.15 and the above decrease by 0.1. The formula is represented again more readably in figure 3.

- If $p_{sum} < 1$, evidently too much load was distributed and q_i must be decreased, where: $q_i \rightarrow q_i * p_{sum}$.

- The processor MP_i distributes load to other processors MP_k if it becomes the case that $q_i > 0.25$.

The method according to the invention thus has the following properties and advantages:

A very small information overhead between the processors participating in the load distribution method. Only a few, preferably three-value, load states are reciprocally known, which load states are updated and distributed only once per control interval.

For each processor there is a quota which is updated in each control interval and regulates the

proportion of the load which is to be distributed from the processor considered to the other processors involved.

For each processor there are individual regulators which divide between the other processors the load that is to be distributed.

The method is not only designed as a "fire-fighting measure" which only takes effect when a processor reaches overload and, if appropriate, tasks (calls) are rejected, rather the load distribution commences earlier and in a softer fashion. As a result, continuous unbalanced load states can be processed better and with fewer rejected tasks (calls).

In the method according to the invention, the load states which are distributed to the other processors are consistently determined on the basis of the estimated offered load and not on the basis of the actually processed load.

The method does not require a load balancing flag which regulates entry into the load distribution. The entry is regulated by way of the distribution quota q_i. Furthermore, mutual dependencies between the load states and the load balancing flag have been eliminated as a result of the absence of a load balancing flag. As a result, the algorithm can more easily be subsequently adapted to changed conditions.

The load-dependent alteration of the individual regulators (load distribution factors p_{ij}) takes place as a function of the number n of processors participating in the load distribution. Consequently, the method is independent of the number of processors involved.

The load-dependent alteration of the distribution quotas and of the individual regulators per control interval takes place in such a way as to avoid excessively slow "creeping" to the optimum value.

The load-dependent alteration of the individual regulators prevents the values from staying at the setting of the preceding load distribution period during a period without load distribution. Rather, there is regulation back to an initial setting.

The inertia - known from the prior art - in the alteration of the quotas has been removed in order to enable easier tracking to the load situation that is actually present.

Attention is supplementarily drawn to the definition of a few terms in this application:
The term or word element "quota" describes the fraction of a whole with a range of values between 0 and 1.

The term of word element "state" describes the instantaneous situation or the instantaneous current value of a quantity. Thus, e.g. the load state of a processor is to be understood as the value of the current load of the processor.

The term hysteresis defines the dependence of a current quantity on its earlier values, its history or its time characteristic.

Patent claims

1. A method for load distribution in a multiprocessor system, in particular in a multiprocessor system of a communication system, in which tasks that arise can be processed by a plurality of processors MP_i (where i = 1,2,...,n) under real-time conditions, having the following iterative method steps that are repeated at time intervals CI:

- each processor MP_i determines its actual current load Y_i and estimates as a function of previously communicated distribution quotas q_i(old) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$, the distribution quota q_i representing the load proportion which can be distributed to other processors MP_k,
- each processor MP_i indirectly or directly communicates its load indication value $MPbi_i$ to the respective other processors MP_k (where k = 1,2,...i-1,i+1,...n),
- each processor MP_i determines its load distribution probabilities p_{ij} (where j = 1,2,...n) as a function of the load indication values $MPbi_k$ of said other processors MP_k,
- each processor MP_i determines its distribution quota q_i(new) as a function of its actual current load Y_i and the load distribution factors p_{ij}

- on the basis of its quota q_i and its load distribution factors p_{ij}, each processor MP_i distributes its distributable load to other processors MP_k if its distribution quota q_i(new) exceeds a predetermined value q_v.

2. The method as claimed in the preceding claim 1, characterized in that the estimated offered load A_i of a processor MP_i is calculated according to the formula $A_i := Y_i / (1 - q_i V)$.

3. The method as claimed in one of the preceding claims, characterized in that the multi-value load indication value (balancing indicator) $MPbi_i$ can assume three discrete values, preferably the values NORMAL, HIGH and OVERLOAD, where NORMAL corresponds to a processor capacity utilization of from 0 to 75%, HIGH corresponds to a processor capacity utilization of from 70% to 85% and OVERLOAD corresponds to a processor capacity utilization of from 85% to 100%.

4. The method as claimed in the preceding claim 3, characterized in that the load indication value (balancing indicator) $MPbi_i$ is subject to a hysteresis with regard to changes.

6. The method as claimed in one of the preceding claims, characterized in that the average or maximum distributable proportion of a typical task CallP is regarded as the typical distributable proportion V.

7. The method as claimed in the preceding claim 6, characterized in that the average or

maximum distributable proportion of a typical task is continually determined as moving average or moving maximum value over a predetermined time period t_D.

8. The method as claimed in the preceding claim 7, characterized in that the following holds true for the predetermined time period t_D: $t_D \gg CI$.

9. The method as claimed in one of the preceding claims, characterized in that an average or maximum task is assumed as the typical task.

10. The method as claimed in the preceding claim 6, characterized in that the average or maximum task is continually determined as moving average or moving maximum value over a predetermined time period t_D.

11. The method as claimed in the preceding claim 10, characterized in that the following holds true for the predetermined time period t_D: $t_D \gg CI$.

12. The method as claimed in one of the preceding claims, characterized in that the following holds true for the predetermined value q_v of the distribution quota q_i starting from which the processor MPi distributes distributable load to other processors MP_k: $0.05<q_v<0.3$, preferably $0.1<q_v<0.25$, preferably $q_v=0.2$.

13. The method as claimed in one of the preceding claims, characterized in that the calculation of the distribution quota q_i satisfies the following criteria:

- p_{ii}: = 0
- if $MPbi_j$ corresponds to an average load, preferably $MPbi_j$ = NORMAL, the following holds true:

  $p_{ij}(new) = pij(old)+p_{c1}/n$, for $j=1,...,$ and $i \neq j$
- if $MPbi_j$ corresponds to a high load, preferably $MPbi_j$=HIGH, the following holds true:

  $p_{ij}(new) = pij(old)-p_{c2}/n$, for $j=1,...,n$ and $i \neq j$
- if $MPbi_j$ corresponds to an overload, preferably $MPbi_j$=OVERLOAD, the following holds true:

  $p_{ij}(new) = 0$
- in which case the $p_{ij\ (j=1,...,n)}$ are normalized to 1 with the sum p_{sum} of the p_{ij} and
- as initialization value at the beginning of the distribution processes, all p_{ij}, excluding p_{ii}, are identical.

14. The method as claimed in the preceding claim 13, characterized in that the following holds true for the constant p_{c1}:

    $0.1<p_{c1}<0.5$, preferably $0.2<p_{c1}<0.3$, preferably $p_{c1}=0.25$.

15. The method as claimed in one of the preceding claims 13-14, characterized in that the following holds true for the constant p_{c2}: $0.1<p_{c2}<0.5$, preferably $0.2<p_{c2}<0.3$, preferably $p_{c2}=0.25$.

16. The method as claimed in one of the preceding claims 13-15, characterized in that the initialization value of the p_{ij} at the beginning of the distribution processes is set to be equal to $(n-1)^{-1}$.

17. The method as claimed in one of the preceding claims 13-16, characterized in that each processor MP_i determines a multi-value load status (load state) $MPls_i$ on the basis of its actual current load Y_i, and the calculation of the load indication values $MPbi_i$ satisfies the following criteria:

- if $MPls_i$ corresponds to the highest load, the following holds true:

  $q_i(new)=c_{q1}$,

- if $p_{sum} \geq 1$ holds true:

  - if the actual current load Y_i is greater than a predetermined value $threshold_H$, q_i is increased where $q_i=\min\{q_i+c_{q1},1\}$,
  - if the actual current load Y_i is less than a predetermined value $threshold_N$, q_i is decreased where $q_i=\max\{q_i-c_{q2}, c_{q3}\}$, where $0<c_{q3}<q_v$, preferably $c_{q3}=0.1$,
  - otherwise ($threshold_N \leq Y_i \leq threshold_H$), qi obtains an intermediate value between the two alternatives mentioned above, preferably by linear interpolation

- if $p_{sum} \leq 1$ holds true: $q_i(new)=q_i(old) * p_{sum}$.

18. The method as claimed in the preceding claim 17, characterized in that the load status (load state) $MPls_i$ is subject to a hysteresis with regard to changes.

19. The method as claimed in one of the preceding claims 17-18, characterized in that the multi-value load status (load state) $MPls_i$ can assume four discrete values, preferably NORMAL (=0 to 0.7), HIGH (=0.7 to 0.85), OVERLOAD (=0.85 to 1) and EXTREME (if load status over a plurality of CI OVERLOAD).

20. The method as claimed in one of the preceding claims 17-19, characterized in that the following holds true for the constant c_{q1}: $0.05<c_{q1}<0.3$, preferably $0.1<c_{q1}<0.2$, preferably $c_{q1}=0.15$.

21. The method as claimed in one of the preceding claims 17-20, characterized in that the following holds true for the constant c_{q2}: $0.05<c_{q2}<0.2$, preferably $c_{q2}=0.10$.

22. The method as claimed in one of the preceding claims 17-21, characterized in that the following holds true for the constant $threshold_N$: $0.6 < threshold_N < 0.8$, preferably $threshold_N = 0.7$.

23. The method as claimed in one of the preceding claims 17-22, characterized in that the following holds true for the constant $threshold_H$: $0.7 < threshold_H < 0.95$, preferably $threshold_H = 0.85$.

24. The method as claimed in one of the preceding claims, characterized in that an overload value OL_i of the processors MP_i is additionally determined for the purpose of quantifying the overload state of the processors, which overload value is a measure of the magnitude of the overload, where $OL_i=0,1,...m$ and the distribution quota q_i is increased in any case if the magnitude of OL_i becomes greater than 0 and $q_i(new):=\min\{qi(old)+c_{q1},1\}$ is set.

25. A multiprocessor system, in particular of a communication system, having a plurality of processors MP_i (where $i=1,2,...,n$) for executing tasks that arise under real-time conditions, in which case:

- each processor MP_i has means for determining its actual current load Y_i and for estimating as a function of previously communicated distribution quotas q_i(old) and the typically distributable proportion V of a typical task its offered load A_i, which leads to a multi-value load indication value (balancing indicator) $MPbi_i$, the distribution quota q_i representing the load proportion which can be distributed to other processors MP_k,

- each processor MP_i has means for indirectly or directly communicating its load indication value $MPbi_i$ to the respective other processors MP_k (where $k = 1,2,...i-1,i+1,...n$),

- each processor MP_i has means for determining its load distribution probabilities p_{ij} (where $j = 1,2,...n$) as a function of the load indication values $MPbi_k$ of said other processors MP_k,

- each processor MP_i has means for determining its distribution quota q_i(new) as a function of its actual current load Y_i and

- each processor MP_i has means for distributing, on the basis of its quota q_i and its load distribution factors p_{ij}, its distributable load to other processors MP_k if its distribution quota q_i(new) exceeds a predetermined value q_v.

26. The multiprocessor system as claimed in claim 25, characterized in that one of the methods as claimed in one of claims 1-24 is implemented.

Dictation A

page 6 of highlighted copy

Each processor MP_i determines a multi-value load status (load state) $MPls_i$ on the basis of its actual current load Y_i, and

Dictation B

page 7

where Ol_i represents a quantification for the overload of the processor,

Page 18

Attention is supplementary drawn to the definition of a few terms in this application:
The term or word element "quota" describes the fraction of a hole with a range of values between 0 and 1.

The term of word element, "states" describes the instantaneous situation or the instantaneous current value of a quantity. Thus, e.g. the load state of a processor is to be understood as the value of the current load of the processor.

The term hysteresis defines the dependence of a current quantity on its earlier values, its history or its time characteristic.

Fig. 1

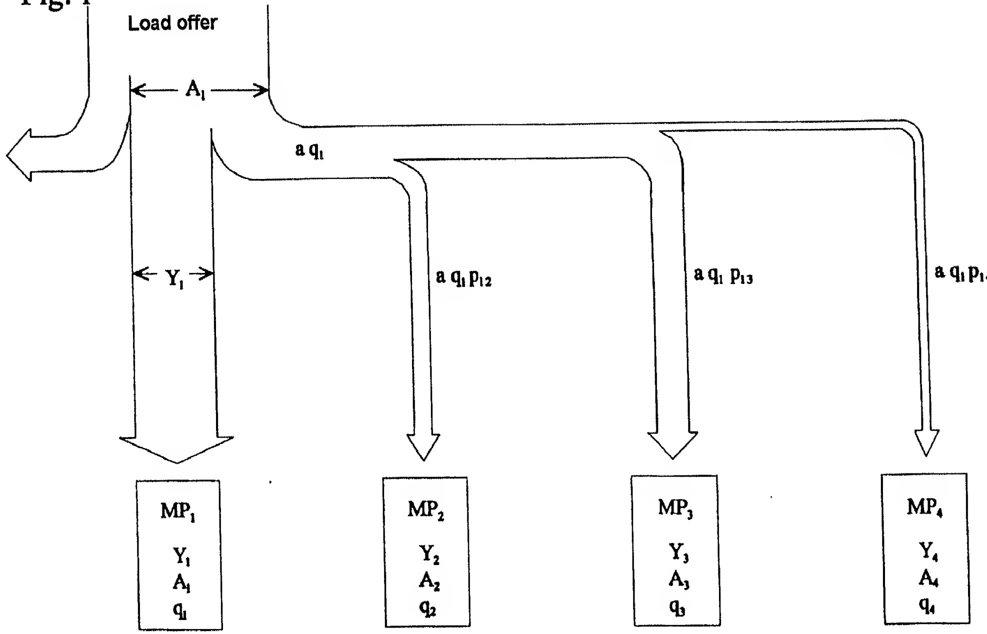
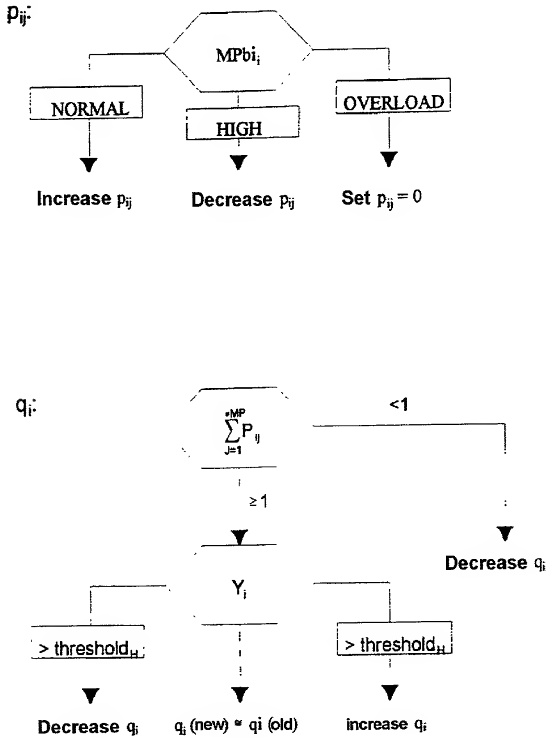

Load offer
A_1
Y_1
$a\,q_1$
$a\,q_1\,p_{12}$
$a\,q_1\,p_{13}$
$a\,q_1\,p_{14}$
MP_1 Y_1 A_1 q_1
MP_2 Y_2 A_2 q_2
MP_3 Y_3 A_3 q_3
MP_4 Y_4 A_4 q_4

Fig. 2



p_{ij}:
$MPbi_i$
NORMAL
HIGH
OVERLOAD
Increase p_{ij}
Decrease p_{ij}
Set $p_{ij} = 0$
q_i:
$\sum_{J=1}^{\#MP} P_{ij}$
<1
≥ 1
Decrease q_i
Y_i
> $threshold_H$
> $threshold_H$
Decrease q_i
q_i (new) = qi (old)
increase q_i

Fig. 3

$$q_i \longrightarrow \min \left\{ \max \left\{ q_i + \frac{0.25}{\text{threshold}_H - \text{threshold}_N} (Y_i - \text{threshold}_N) - 0.1,\ 0.1 \right\},\ 1.0 \right\}$$

Declaration and Power of Attorney For Patent Application
Erklärung Für Patentanmeldungen Mit Vollmacht
German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

Lastverteilungsverfahren eines Multiprozessorsystems und Multiprozessorsystem

deren Beschreibung

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am 17.01.2000 als
PCT internationale Anmeldung
PCT Anmeldungsnummer PCT/EP00/00317
eingereicht wurde und am ________
abgeändert wurde (falls tatsächlich abgeändert).

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Load distribution method for a multiprocessor system and corresponding multiprocessor system

the specification of which

(check one)
☐ is attached hereto.
☒ was filed on 17.01.2000 as
PCT international application
PCT Application No. PCT/EP00/00317
and was amended on ________
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Form PTO-FB-240 (8-83)
Patent and Trademark Office-U.S. DEPARTMENT OF COMMERCE
IDNR: 2590 / V: 99-1.00 / B:Val

German Language Declaration

Prior foreign appplications
Priorität beansprucht

			Priority Claimed	
99101122.2	EP	21.01.1999	☒	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein
			☐	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein
			☐	☐
(Number)	(Country)	(Day Month Year Filed)	Yes	No
(Nummer)	(Land)	(Tag Monat Jahr eingereicht)	Ja	Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

PCT/EP00/00317	17.01.2000	anhängig	pending
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D, M, Y) (Anmeldedatum T, M, J)	(Status) (patentiert, anhängig, aufgegeben)	(Status) (patented, pending, abandoned)
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D,M,Y) (Anmeldedatum T, M; J)	(Status) (patentiert, anhängig, aufgeben)	(Status) (patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*	POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*
	And I hereby appoint
Customer No.	
Telefongespräche bitte richten an: *(Name und Telefonnummer)*	Direct Telephone Calls to: *(name and telephone number)* Ext. ________________
Postanschrift:	Send Correspondence to:

Bell, Boyd & Lloyd LLC
Three First National Plaza, 70 West Madison Street, Suite 3300 60602-4207 Chicago, Illinois
Telephone: (001) 312 372 11 21 and Facsimile (001) 312 372 20 98
or
Customer No.

Voller Name des einzigen oder ursprünglichen Erfinders: PETER HANSELKA 1-00	Full name of sole or first inventor: PETER HANSELKA
Unterschrift des Erfinders: Peter Hanselka — Datum: 4.7.01	Inventor's signature: Peter Hanselka — Date: 4.7.01
Wohnsitz: MUENCHEN, DEUTSCHLAND	Residence: MUENCHEN, GERMANY DEX
Staatsangehörigkeit: DE	Citizenship: DE
Postanschrift: PIDINGER STR. 13	Post Office Addess: PIDINGER STR. 13
81379 MUENCHEN	81379 MUENCHEN
Voller Name des zweiten Miterfinders (falls zutreffend): GABRIELE RAICHLE 2-00	Full name of second joint inventor, if any: GABRIELE RAICHLE
Unterschrift des Erfinders: G. Raichle — Datum: 2.7.01	Second Inventor's signature: Gabriele Raichle — Date: 2.7.01
Wohnsitz: MUENCHEN, DEUTSCHLAND	Residence: MUENCHEN, GERMANY DEX
Staatsangehörigkeit: DE	Citizenship: DE
Postanschrift: AMBACHERSTR. 39	Post Office Address: AMBACHERSTR. 39
81476 MUENCHEN	81476 MUENCHEN

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

Form PTO-FB-240 (8-83)

Patent and Trademark Office-U.S. Department of COMMERCE

over

Voller Name des dritten Miterfinders:	Full name of third joint inventor:
Dr. CHRISTOPHER ROSS 3 - oo	Dr. CHRISTOPHER ROSS
Unterschrift des Erfinders / Datum: Chris Ross 27.7.01	Inventor's signature / Date: Chris Ross 27th July 2001
Wohnsitz: MUENCHEN, DEUTSCHLAND	Residence: MUENCHEN, GERMANY DEX
Staatsangehörigkeit: DE	Citizenship: DE
Postanschrift: KIRCHENSTR. 89	Post Office Address: KIRCHENSTR. 89
81675 MUENCHEN	81675 MUENCHEN

Voller Name des vierten Miterfinders:	Full name of fourth joint inventor:
Unterschrift des Erfinders / Datum	Inventor's signature / Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des fünften Miterfinders:	Full name of fifth joint inventor:
Unterschrift des Erfinders / Datum	Inventor's signature / Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des sechsten Miterfinders:	Full name of sixth joint inventor:
Unterschrift des Erfinders / Datum	Inventor's signature / Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).